

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

昭63-127334

⑬ Int.Cl.⁴

G 06 F 9/06
12/14
15/16

識別記号

3 3 0
3 2 0
3 2 0
4 2 0

庁内整理番号

A-7361-5B
F-7737-5B
P-2116-5B
S-2116-5B

⑭ 公開 昭和63年(1988)5月31日

審査請求 未請求 発明の数 2 (全36頁)

⑮ 発明の名称 ソフトウェア保護機構から安全に実行権を取出し、及び条件付けする方法

⑯ 特 願 昭62-250063

⑰ 出 願 昭62(1987)10月5日

優先権主張 ⑱ 1686年11月5日 ⑲ 米国(US) ⑳ 927299

㉑ 発 明 者 リーム・デヴィッド・カ アメリカ合衆国ニューヨーク州カーメル、ヴァリエー・ロード、ロード1、ボックス191番地

㉒ 発 明 者 ステイブ・リチャード・ホワイト アメリカ合衆国ニューヨーク州ニューヨーク、アパートメント33、パーク・アヴェニュー7番地

㉓ 出 願 人 インターナショナル・ビジネス・マシーンズ・コーポレーション アメリカ合衆国10504、ニューヨーク州 アーモンク(番地なし)

㉔ 代 理 人 弁理士 山本 仁朗 外1名

明 細 書

1 発明の名称 ソフトウェア保護機構から安全に実行権を取出し、及び条件付けする方法

2. 特許請求の範囲

(1) 上位計算機に関連する論理的に安全な、少なくとも第1のキー及び特定のアプリケーションを実行する権利を喪失す第2のソフトウェア・キーを記憶するコプロセッサを有し、上記上位計算機が上記第2のソフトウェア・キーによつて暗号化キーの下に暗号化された上記特定のアプリケーションにアクセスできるソフトウェア保護機構の上記コプロセッサから実行権を取出すための方法であつて、

(a) 上記コプロセッサに、少なくとも書き込み可能な媒体及び物理的及び論理的に安全な、平文のトークン・データを記憶する媒体を与え、

(b) 上記コプロセッサに上記第1のキーで暗号化された上記平文のトークン・データより成るデ

ータ・ブロックを与え、

(c) 上記データ・ブロックを上記コプロセッサ中で解読して、上記平文トークン・データを発生し、

(d) 上記平文のトークン・データを上記第2のソフトウェア・キーで暗号化して、対応するデータ・ブロックを発生し、

(e) 上記ソフトウェア・キーを上記第1のキーによつて暗号化して暗号化したソフトウェア・キーを発生し、

(f) 上記対応するデータ・ブロック、暗号化したアプリケーション及び上記暗号化したソフトウェア・キーを上記転送セットに書き込んで、上記ソフトウェア・キーを上記コプロセッサから抹消し、

以て上記ソフトウェア・キーを上記コプロセッサから除去して上記転送セットに書き込む段階を有する、

ソフトウェア保護機構から実行権を安全に取出す方法。

(2) ソフトウェア・キーによつて暗号化された部

分を少なくとも含む特定の保護されたアプリケーションを上位プロセッサ並びに物理的及び論理的に安全なコプロセッサを含む複合計算機上で、実行するための、上記コプロセッサの永久メモリ中に記憶されている上記ソフトウェア・キーによつて表わされた実行権に条件を付する方法であつて、

(a) 上記コプロセッサに少なくとも1つの条件のステートメントを与え、

(b) 上記永久メモリ中に、上記ソフトウェア・キーの記憶とともに、上記条件に関連するデータを記憶し、

(c) 上記ソフトウェア・キーもしくは上記保護アプリケーションの使用を許可する前に上記ステートメント及び上記データにアクセスすることを上記コプロセッサに要求し、

(d) さらに上記コプロセッサに上記ステートメント及び上記データを比較し、上記条件が満足されているかどうかを要求し、上記条件を満足している時にのみ使用を許可する段階を有する、

保護されたアプリケーションの実行権に条件を

付ける方法。

3. 発明の詳細な説明

以下の順序で本発明を説明する。

A 産業上の利用分野

B 従来技術

C 発明が解決しようとする問題点

D 問題点を解決するための手段

E 実施例

E 1 ソフトウェア資産保護機構(第2図)

E 2 実行権の制約(第1図、第3図、第4図)

E 3 実行権の転送(第5図、第6図、第7図、第19図)

E 4 実行権のバックアップ(第8図、第9図、第10図、第11図、第12図、第13図、第14図、第15図、第16図、第17図)

E 5 販売者のキーの暗号化(EVK)(第18図)

E 6 ソフトウェアの返品

F 発明の効果

A 産業上の利用分野

本発明はデータ処理、具体的にはソフトウェア保護機構に関する。この機構は磁気媒体もしくは他の媒体によつて頒布されるソフトウェアを許可された物理的に安全なコプロセッサに関連する任意の計算機上で使用するように制限する。この機構はユーザのバックアップ・コピーの作成を防げないが、その保護はこのようなバックアップ・コピーによつて危くされることはない。本発明は特にコピー防止機構の顕著な特徴である実行権の処遇に関する。

B 従来技術

関連文献としては、本出願人に係る次の米国特許出願明細書があげられる。

米国特許出願第927309号(1986年1月5日出願)

米国特許出願第927306号(1986年1

1月5日出願)

米国特許出願第927629号(1986年1

1月5日出願)

米国特許出願第927298号(1986年1

1月5日出願)

米国特許出願第927286号(1986年1

1月5日出願)

米国特許出願第927297号(1986年1

1月5日出願)

基本的なコピー防止機構は上述の米国特許出願第927629号に開示されている。この機構は保護されるべきソフトウェアをこのソフトウェアの実行権から分離している。安全を与え、上記機構を具体化するために、保護されたアプリケーションを実行する各計算機(以下上位計算機と呼ぶ)は論理的及び物理的に安全なコプロセッサと関連付けられる。コプロセッサ中に設置される時は、特定の保護されたアプリケーションの実行権はアプリケーション・キー(AK)と呼ばれるソフトウェア解読キーの形で存在する。ソフトウェア解

読キーAKがコプロセッサの永久メモリに保持されている限り、これに対応する保護されたソフトウェアは上位計算機及びコプロセッサを含む複合システム上で実行できる。このソフトウェアのコピー防止機構は現存の及び計画中のソフトウェア頒布技術にほとんど抵触せず、ユーザに無制限のバックアップ(補助)コピーの作成を可能にし、ユーザとソフトウェアの販売者間に2方向通信を必要としないという利点がある。この利点はハードウェア・カートリッジ(即ちトークン)の形で与えられる、実行権を受取るための許可をコプロセッサに頒布することによつて支援される。さらにユーザは未使用のトークンによつて表わされる実行権をコプロセッサに転送するため、保護されたアプリケーションを実行する始めにトークンを使用するだけでよい。その後、トークンは廃棄され、その後はこの実行権(トークン)の保持もしくは使用は全く必要がない。

上記米国特許出願第927629号に開示されている発明は(ユーザが最初どのようにして実行

した時の頒布媒体と同じである)の使用によつて間接的か、コプロセッサからコプロセッサへの通信リンクを介して直接的に行うことができる。尚、このとき、転送トランザクションが傍受されるかも知れないという意味で通信が安全でない時も安全が保持される。

本発明は又実行権に条件を付けるための技術を与える。たとえば実行権は期間(べ切り日もしくはべ切り時間迄存在する権利)によつて条件を付けられなくてはならず、又呼び起される回数によつて条件を付けることができる(たとえば販売者は保護されたアプリケーションを10回実行する権利をユーザに販売することができる)。以下説明するように実行権は実行権のソース(ソフトウェアの販売者)を満足するようにコプロセッサによつて測定できる限り任意の他のパラメータによつて条件を付けることができる。条件を付された実行権が使用されると、ソフトウェアの販売者にさらに柔軟性が与えられ、ソフトウェアの分野に始めて真に安全な「返品」対策の可能性が開かれ

権を獲得するかを説明しているだけであり)実行権の処遇については向けられていず、実行権に条件を付ける可能性についての説明がない。

C 発明が解決しようとする問題点

本発明の目的はコプロセッサ中に存在するソフトウェアの実行権に条件を付け、その処遇を決め、もしくはこれを転送することにある。

本発明の他の目的は、実行権を安全に転送する機能を与えることにある。

D 問題点を解決するための手段

ソフトウェアの実行権は他のコプロセッサに転送できもしくは、外部に記憶するためにコプロセッサの外部に移すことができる。いずれの場合にも、実行権を転送する過程は擬似的もしくは複製の実行権を発生したり、許したりするものでないことが重要である。このようなことは勿論コピー防止の目的に反する。以下説明するように、実行権の転送は転送セット(多くの点で実行権を獲得

る。明らかに、現在のソフトウェアの頒布技術を使用するソフトウェアの販売者は完全購買信用取引のソフトウェアの返品を受取るようなことがあれば彼の製品を只で与えるという危険がある。現在の頒布技術では、販売者はユーザがすでにソフトウェアを複製して返品した時に、ユーザが完全に使用可能なアプリケーションのコピーを保持しているかどうかを検証する手段がない。本明細書の原理を使用すると、ソフトウェアの販売者は「返品」対策を具体化でき、ユーザがソフトウェアを返品すると、ユーザはもはや実行可能なコピーを保持できなくなる。

ソフトウェアのコピー防止機構は現実、現実の装置により動作するためと、別個の実行権がコプロセッサの永久メモリ中に暗号キーの形で存在するので、実行権を記憶するコプロセッサが故障する可能性も考慮しなければならない。このような故障が生じてユーザの実行権が完全に失われるようなことがあつてはならない。本発明はユーザのコプロセッサが故障した場合における実行権

の消失に対してユーザを保護する装置及び方法を与える。実行権を移動即ち転送する場合と同じように、任意のハードウェアのバックアップ技術（コプロセッサの故障の場合に利用可能になる）は擬似実行権を許容する性質を持つていてはならない。

本発明のハードウェア・バックアップ方法は実行権を不当に複製する機会を最小にし、その活動を妨げる。

条件の付された実行権

実行権に条件を付すためには、上記米国特許出願第927629号に開示されているシステムに、次のものがなくてはならない。

- (1) アプリケーション・ソフトウェアの実行が完全に許される（もしくは許されない）条件のステートメント
- (2) この条件を測定できるある客観的規準
- (3) 条件と規準を比較して、比較の結果によつて実行されるソフトウェア・プログラム

これ等の目的はユーザもしくはソフトウェアの

憶スペースはすでに暗号化ソフトウェアを解説するのに必要な解説キーを記憶する機能が割当てられている。従つてソフトウェアの特定の保護された部分に割当てられている記憶スペースを拡張して規準とテストされる条件を含めるようにする。メモリが持久性であるために実行権がコプロセッサ中に利用可能であるかぎり、この客観的條件も利用可能である。コプロセッサはその物理的に安全な境界内に連続的に電力が供給される実時間クロックを含むので、時間を含む規準を使用したい時は、時間情報が利用可能である。この情報はコプロセッサの持久メモリ中に記憶され、特定のアプリケーションに割当てられたこのメモリの部分だけがそのアプリケーションによつてアクセスできるので、情報はユーザによる修正から安全に保護される。アプリケーション・ソフトウェアは持久メモリのこの部分に記憶された条件を変更できるかもしれないが、実時間クロックの値は変更できない。

たとえば、ソフトウェアはこのメモリ中に保持

販売者によつて特別に許可されていない他人がこの条件もしくはこの条件の満足をテストする客観的規準を変えようとする試みに対して安全な方法で満足されなければならない。

本発明に従い、この規準はソフトウェア中に、より具体的にはアプリケーション・ソフトウェアの保護された、即ち暗号化部分中に記述される。上記米国特許出願第927629号に説明されているように保護されたアプリケーション・ソフトウェアがユーザに使用される唯一の形式は暗号形式においてである。それはユーザはデータ・オブジェクトとして暗号キーにアクセスできず、保護されたソフトウェアを修正したり、読取りすらできないからである。保護ソフトウェア内に実行権の条件を組み込むことによつて、ソフトウェアの販売者によつて許可されない限りこれ等の条件はユーザもしくは他人による変更から安全に保護される。プログラムされた規準に対してテストする条件を保管するためには、コプロセッサ中の持久メモリ中のある記憶スペースを使用する。この記

している回数を変更することによつて回数もしくは総時間がカウントでき、実行の回数もしくは総時間に関連する規準がもはや記憶した条件を満足しなくなる迄実行できる。

たとえば、ソフトウェアの販売者がユーザにある期限の日付を超えてはならないという条件付き（たとえば、ユーザは1987年の3月1日迄は保護されたアプリケーションを実行できるが、その後は不可という権利を有する場合）で実行権を転送したものと仮定する。従つてコプロセッサの動作命令はソフトウェア解説キーとともに最後の許可日付（期限の日付）が記憶できなくてはならない。コプロセッサは実時間クロックを保持しているので、解説キーをアクセスする時もしくはアプリケーションの実行の途中にはいつでも、期限の日付及び現在の日付が利用できる。この期限の日付けは実時間クロックのセット時間と同じようにコプロセッサの安全性によつて不法な変更から守られている。ソフトウェアの暗号化部分（保護部分）は実行が期限の日付けを越えていると使用

できないという規準を記述している。保護ソフトウェアが実行される時はいつでも、解説キーと期限の日付けがコプロセッサの持久メモリからアクセスされる。保護ソフトウェア中でテストされる規準は期限の日付を現在の日付と比較することを必要とする。もし現在の日付が期限の日付を越えていると、保護ソフトウェアの実行は進行しない。保護ソフトウェアは又現在の日付が期限の日付を越えると特定のソフトウェア解説キーを抹消できるように作成できる。この分野の専門家にとつては期限の日付に代る条件としてソフトウェアを実行する回数を使用できることは明らかであろう。この場合、保護ソフトウェアはソフトウェア解説キーとともに現在のデータを記憶する代りに、許可された実行回数を記述し、許容使用回数を記憶していて、この回数をソフトウェアの実行のたびにデクレメントする。従つて保護部分が許容実行回数を規準と比較して、残りの回数が0より大きいことを確かめる。許容実行回数が0になると、ユーザのソフトウェア実行要求は拒否される(お

造者によつてロードされる。ユーザによつて使用される前の転送セットは2つの形で記憶される単一の情報片であるトークン・データを有する。トークンはハードウェアの販売者によつて平文のトークン・データとしてロードされる。トークンの物理的特性がこの高度に機密性の情報を不許可の者から保護する。同じデータが共通スーパーバイザ・キー(CSK)と呼ばれるハードウェアの製造者の秘密のキーによつて暗号化され E_{CSK} (トークン・データ)が発生される。これは転送セットのディスク上もしくは(可能な構造になつていれば)トークン中に記憶される。 E_{CSK} (トークン・データ)は暗号化されているから、誰もが見ることができ、さらにコピーできるようなディスク上に記憶できる。転送セットはハードウェアの販売者のような信頼の置けるソースによつて製造されることが必要である。それは仮にトークンの内容が知られると、他のトークンにその知られた内容がロードでき、転送される実行権が複製されるからである。ユーザが適切な転送セット

そらくソフトウェア解説キーも抹消される)。このような特定の具体例には経過時間、パスワード及びこれ等の組合せ並びに他の測定可能な変数を含む多くの変形が存在することは明らかであろう。

実行権の転送

1人のユーザから他のユーザへの実行権の転送(具体的にはソース・コプロセッサからシンク・コプロセッサへの転送)は頒布セットを再構成することによつて達成できる。この手順は最初に実行権を獲得した形と実質的に同じであるポータブルな形に戻す(上記米国特許出願第927629号参照)。この手順は必然的に実行権をソース・コプロセッサから移動させる。

このトランザクションはユーザがトークン(一種の割印)の構造に依存してトークンもしくはディスクのいずれか及びトークン対(転送セットとも呼ばれる)を得ることを必要とする。これ等のセットはハードウェアの販売者によつて与えることができる。これ等のセット中のトークン(即ちカートリッジ)はコプロセッサのハードウェア製

を獲得したと仮定すると、頒布セットはたとえば次のようにしてユーザと彼の複合計算システムによつて、頒布セット再構成(BDS)過程を使用して作成される。

上位計算機上で実行中のユーティリティ・プログラムが(ソース)コプロセッサにBDSシーケンスを開始することを知らせる。ユーティリティ・プログラムはコプロセッサに転送すべきキーの位置を知らせる。コプロセッサはインデックス付きキーを除くすべての許可されたキーについてCBS(バックアップ・セット作成)手順を実行する。CBS手順については後に説明する。この時点では、CBS手順は存在する任意のハードウェア・バックアップ機構を無効にすると述べておくだけで十分である。コプロセッサは転送セットから暗号化されたトークン記述子 E_{CSK} (トークン・データ)のコピーを要求してこれを受取る。コプロセッサはトークン記述子を解説して平文のトークン・データを与える。この平文トークン・データは次にインデックスによつて識別されるソ

ソフトウェア解読キーを使用して暗号化され、 E_{AK} (トークン・データ)を発生する。次にコプロセッサはこの暗号化トークン記述子 E_{AK} (トークン・データ)をトークンもしくはディスク上の予約された持久メモリ領域中に記憶し、所定のメモリ位置にあるソフトウェア解読キー A_K を抹消即ち非活性化する。従つてコプロセッサは暗号化トークン記述子を上位計算機に渡して伝送セット・ディスク上に記憶する。以下説明するように、伝送すべきキー (A_K) は実行の条件に関連付けられる。もしこれ等の条件が(期限の日付のように)不変ならば、暗号化アプリケーション・キーは伝送セット・ディスクにコピーできる。もし実行の条件が変わると(使用の残り時間もしくは使用の残り回数のように)、アプリケーション・キーを含む暗号化ファイル及び実行の条件はこの条件をリセットすることなくしては頒布ディスクからのコピーができない。トークン記述子ファイルとアプリケーション・キー・ファイルのこの同期は各ファイル中に照応テスト回数を含ませることによつ

クアップ・コピーからコピーできる。頒布セットは一つの点を除いてすべての点で同じであるから、これは任意のコプロセッサによつて特定の暗号化ソフトウェア・ファイルを実行する権利を伝送するように使用可能である。この頒布セットと実行権を獲得した原頒布セット間の唯一の差異はトークン・データが異なっている可能性が極めて大きいことである。しかしながらこのことはセットの両方の部分(カートリッジ及びディスクセット)が同一のトークン・データから得られる限り重要ではない。(ソース)コプロセッサはここで一時メモリから解読キー A_K を抹消する。この時点で実行権はソース・コプロセッサから抹消され、伝送セットだけに存在する。

しかしながら、実行権の伝送は間接的、たとえば上述のように伝送セットを介して行う必要はない。実行権の伝送は直接的でよく、コプロセッサからコプロセッサへの通信リンクを使用しても行うことができる。さらにコプロセッサ間の通信リンクは伝送される実行権の安全を守るために保護

て達成できる。伝送の次の段階で伝送ディスク上に記憶させるために暗号化アプリケーション・ファイルを作成する。この作成は照応テスト回数が乱数に代る点を除き以下に説明する販売者キー暗号化 (E_{VK}) と同じである。この照応数はトークン・データの一部であつてよい。この作成と伝送の後に、上位計算機中で実行されているユーティリティ・プログラムは伝送セット・ディスクに保護されたプログラムの平文と暗号文の部分を含む2つのファイルを伝送する。ここでこの頒布セットはこれがソフトウェア解読キー A_K で暗号化された暗号化トークン・データ、平文ファイル及び同じくソフトウェア解読キー A_K で暗号化された保護された即ち暗号化ソフトウェア・ファイル並びにハードウェアの販売者のキー C_{SK} によつて暗号化されたソフトウェア解読キーを含む点上記米国特許第927629号に開示されている頒布セットと同じことがわかるであろう。後の3つの要素はもし実行条件が許容されているならば原頒布ディスクもしくはこれ等のファイルのパッ

されている必要はなく、安全は以下説明するように暗号化によつて保護される。以下の説明ではコプロセッサ間の伝送について説明するが、コプロセッサ間の通信リンクは直接でもよく、双方向データ通信システムを介するものでもよい。1つのコプロセッサから他のコプロセッサへの実行権の伝送は関与する2つのコプロセッサが互を「ファミリのメンバ」として確認でき、このトランザクションに使用するための一回限りのセッション・キーを発生できるときは安全と考えることができる。ファミリのメンバであることの識別は各々の使用する手順が正しくかみ合い、従つて実行権の拡散が生ぜず、他の保護された情報が露顕しないことを保証するために必要である。

セッション・キーを発生するトランザクションはこのトランザクションに関与する2つのコプロセッサ・システムに共通な高い特権メモリ中に情報が存在すること、及びコプロセッサが良い乱数を発生することができる能力に依存する。セッション・キーを発生する過程は両コプロセッサが必

要な前提情報を共通に保持している限り、トランザクションに關与する両コプロセッサに同じキーを所有させることができる。

相互識別とセッション・キーの発生のトランザクションは次のように行われる。説明の目的のために、実行権があるユーザ、複合計算システムもしくはコプロセッサから転送される時はこのユーザ、ユーザの複合計算システムもしくはユーザのコプロセッサをソース・ユーザ、ソース複合計算システムもしくはソース・コプロセッサと呼ぶことにする。実行権が転送されて来る時は、そのユーザ、複合計算システムもしくはコプロセッサをシンク・ユーザ、シンク複合計算システムもしくはシンク・コプロセッサと呼ぶことにする。ソース・ユーザはソース・コプロセッサにセッション・キーが必要になったことを信号する。ソース・コプロセッサはセッション・キーを発生するのに使用する乱数を発生し、これを他のランダムなビット・ストリングに埋込み、メッセージ確認コード(MAC)を添付する。MACは解説によつて

復元した平文メッセージが転送されて来たメッセージに同じであることを保証するのに使用できる。ソース・コプロセッサはこの数をキーCSKで暗号化し、暗号化した数をシンク・コプロセッサに送る。シンク・コプロセッサも同じ機能を実行してその暗号化乱数をソース・コプロセッサに送る。ソース・コプロセッサは暗号化キー(CSK)によつて受取った乱数を解説する。各コプロセッサが多くのCSKを記憶している場合には各コプロセッサは受取った数を各CSKによつて相繼いで、有効なMACが得られるか、スーパーバイザ・キー(CSK)のコレクションが尽きる迄解説する。有効なMACが発見されなかつた時は、誤りメッセージが戻される。この誤りメッセージの発生はファミリのメンバーでないコプロセッサがそうであるものとしてそれ自身を偽装した場合の代表的な結果である。もし有効なMACを得た場合には、2つのコプロセッサ中で発生した乱数は両コプロセッサ中で組合され、セッション・キーとして使用される。勿論コプロセッサのための動作命令の一

部は乱数を組合して、セッション・キーを発生する特定の方法を前もつて決定している。ここで、1つの乱数は他の乱数と連結でき、もしくは排他的にORできる等々である。

正しいスーパーバイザ・キーの検索において相続く解説を必要とする手順を避けるために、使用したスーパーバイザ・キーのためのインデックスのような位置情報を暗号化した数とともに送ることができる。しかしながら、コプロセッサに正しいスーパーバイザ・キーを探査させることを要求する技術には長所がある。暗号化した数とともにインデックスされた数を送る手順は傍受者に対してスーパーバイザ・キーに関する情報の集まりを増大させる。

セッション・キーが一度発生すると、コプロセッサは実行権をセッション・キーで暗号化して転送することが可能になる。この手順を制御するコプロセッサのファームウェアは実行権がソース・コプロセッサから抹消されることを保証する。実行権はソース・コプロセッサによつて転送される時に

抹消できるが、好ましい技術はシンク・コプロセッサが実行権を安全に受取った時にのみ実行権(ソフトウェア・キー)を抹消すること及びソース・コプロセッサが実行権が非活性化されたことを示す時にのみ実行権をアクティブにするものである。暗号化されて転送される実行権は、その暗号化によつて傍受者から安全であり、安全な通信リンクもしくはチャネルを使用する必要はない。

直接、即ちコプロセッサ間通信は回路網もしくは本体リンク環境で実行権を移動する方法である。これ等の場合においてもしくは特定のコプロセッサ中の特定のアプリケーション中に2つ以上の権利を設定することを含む他の場合には、所与のパッケージ当りの権利のカウントがソース・コプロセッサのキー記憶領域中に保持され、すでに受取られている権利の数が頒布されて来る権利を制限するようにされる。

上述のことから、AKをいくつかの次元(時間、使用回数等)のAKの寿命の記述に關連付けるの

はこの機構の領域内であることは明らかであろう。

この機構の能力にはソフトウェア販売者によつて与えられ、コプロセッサによつて実行されるソフトウェアの制御の下にこの寿命を分割し、区分することが含まれることが明らかであろう。ソフトウェアのライブラリは実行権の保存を侵害することなく、分散計算システムのユーザに利益を与えるこれ等の手段によつて分散計算システム中に保持できる。

複数の実行権の転送を必要とする場合には、コプロセッサ間の転送は次のように一度セッション・キーが作成されてから行われる。ソース・ユーザはソース・コプロセッサに対しこれ等の実行権(AK)を転送させることを確認させる。ソース・ユーザはCBS手順(以下定義する)を実行し、実行権のバックアップ・セットを更新する。次にソース・コプロセッサはセッション・キーによつてソフトウェア・キーを暗号化し、これ等をセッション・キーとともに一時メモリの予約された位置中に記憶する。ソース・コプロセッサは各ソフ

トウェア・キーをキー・メモリ(永久メモリ)に非活性としてマークし、暗号化情報をシンク・コプロセッサに送る。シンク・コプロセッサは暗号化キーの集りをソース・コプロセッサから受取り、これ等をセッション・キーによつて解読する。シンク・コプロセッサは次にキー・メモリ(永久メモリ)に解読したキーを記憶し、これ等を非活性とマークし、その上位計算機に位置情報を送る。この位置情報によつてアプリケーションはロード解読実行手順でキーをアクセスできる。実際に保護されたソフトウェアより成るファイルは通常の(非保護の)技術もしくは装置によつて送ることができる。シンク・コプロセッサはソース・コプロセッサにメッセージを戻して特定のソフトウェア・キー(単、複)の受取りを示して、ソース・コプロセッサにその一時メモリからこれ等のソフトウェア・キーを除去せしめる。ソース・コプロセッサがこの除去を確認すると、キーがシンク・コプロセッサによつて活性化される。

ソフトウェア販売者は実行権の転送が可能なの

を欲しないかも知れず、それによつて販売の条項もしくは条件が定められることに注意されたい。実行権の許容可能な移動を制御する条件はAKを獲得した手順の1部によつてAKと関連して記憶できる。

AKの処遇がユーザによつて要求されると、これ等の条件(フラグ)は、要求された処遇を具体化したコプロセッサ・ファームウェア中に記憶された進行/非進行規準とテストされる。

実行権のバックアップ

以下説明するバックアップ手順はユーザのコプロセッサ中に記憶した実行権の集りが、不測の故障、たとえば、コプロセッサの持久メモリの電源の故障によつて失われないことを保証するものである。このAKのバックアップ手順はデータ及びソフトウェアのファイルに適用されるバックアップ・手順とは区別すべきものである。後者の手順は完全に通常のもので、一般に知られているものであり、このシステム中に存在する時は、これ等の種類のオブジェクト(平文及び暗号化したソフ

トウェア並びに暗号化したアプリケーション・キー)に適用できるものである。これ等のオブジェクトの多くの機能的コピーは任意の認可システムによつて使用するために保存され、損失に対処できるようになつている。これに対し、以下説明するバックアップ手順の目的は実行権をバックアップして、これ等が失われることなく、この種のオブジェクトの多くの機能的コピーが作成できないようにすることにある。

このバックアップ手順はコプロセッサ中の不測の故障の影響を免がれるように設計されているので、このバックアップ手順はプロセッサの外部で十分な情報を作成できて、完全に有効な実行権を記憶する(ソース)プロセッサとは完全に独立に実行権設置できるものでなくてはならない。このような独立性のために、バックアップ手順自体が(複製もしくは擬似的の)実行権の可能なソースになる。バックアップ権利の作成の許可はソフトウェア販売者の施策決定による。このバックアップ権利の作成はソフトウェアの販売者に損害を与

える可能性があるが、カスタマの便宜のために与えられるものである。権利の転送に関してすでに説明したように、ソフトウェアの供給者はバックアップを許可することを望まないであろう。この販売の条件はA Kに関連する非バックアップ・フラグが存在すると、転送の制御について説明したのと同じ機構によつて強化できる。このオプションは実行権が変更できるような場合に望ましい。バックアップ権の影響を最小にするために、これ等はこのシステム中では条件付きにされ、即ち実際に故障したコプロセッサを有するユーザが例えばハードウェアの製造者によつてこの故障を検査させるに十分長い、適当な短い時間間隔の後に切れるようにされる。ハードウェア製造者がユーザのコプロセッサの故障を検査したと仮定すると、ユーザにバックアップ権に与えられた条件を除去する追加の手順を遂行する手段が与えられる。バックアップ権の発生と、以下説明する他の能力とを安全に統合するために、バックアップ権の発生は最初はこの時未完のまま残されているすべての

に、設置された実行権の縮小したセットを正しく反映したバックアップ・セットを転送後に直ちに形成することもできる。

バックアップ権を発生するための手順は単一のバックアップ・セットが多くのアプリケーションのために潜在実行権を保持できる点で包括的である。この手順はユーザの実行権の集りが動的であること、たとえばある日の実行権の集りが他の日の実行権の集りと比較して大きいかもしれないことを考慮している。これによつてバックアップ権も動的であり、ユーザの実行権の集りの変更を追跡することが要求される。

基本的には、このバックアップ手順は2段階過程であり、第1段階では、バックアップ・セットが始めて発生される(バックアップ・セット作成、即ちC B S)。バックアップ・セットは(一群のアプリケーションのための)実行権の暗号化ファイルを記憶するディスクを含む。この(ソース)プロセッサに一意的なスーパーバイザ・キー(U S K)も又このファイル中に含まれる。この情報

許容された権利の転送であると考えらるべきである。転送手順は必然的にバックアップ手順と関連付けられ(可能な実行権を保持している)ユーザのバックアップ・セットは完全な転送が行われる時に無効にされる。これによつて転送された権利が同じように有効なバックアップ中に存在するのが防止される。従つて、ユーザはバックアップ手続と関連して次のオプションを有する。

- (1) ユーザの実行権がコプロセッサ上に設置される時は、ユーザは実際の、但し条件の付された実行権に変換できるバックアップ・セット(潜在的な実行権の集りを表わす)を持つことができるか、
- (2) ユーザは1乃至それ以上の実行権を彼のコプロセッサから移動してこれ等を転送セット中に設置するか、もしくは直接他のコプロセッサに転送できるが、この手順は現存のバックアップ・セットの無効化を必要とする。この条件はユーザの権利と完全に両立する。それは実行権がコプロセッサから除去される時は、権利はバックアップ・セットに表わされる必要がないからである。明らか

はハードウェアの製造者が故障の証拠として受取つた時に、故障したプロセッサがこれ等の権利を含むことを主張できるプロセッサであることを検査させるものである。この集りをコプロセッサに対して有効にするトークンも又このセットの一部である。このファイルを暗号化するのに使用できるキーはコプロセッサによつて、たぶん日付及び時間を一意的なスーパーバイザ・キー(U S K)によつて暗号化することによつて発生される乱数である。バックアップ・セットのトークンによつて有効にされるのはこの乱数キー(R K)の使用に対する許可である。上述の如く、バックアップ・セットは保留中の転送と考えることができる。バックアップ過程の第2の段階はバックアップ・セット設置(I B S)である。この段階ではトークン及びディスクより成るバックアップ・セットによつて表わされる潜在実行権がコプロセッサ上に設置される。ユーザはI B S段階を実行する前の複数の機会にバックアップ手順のC B S段階を実行する可能性があると考えているので、バック

アップ・セットに使用されるトークンは頒布セットに使用するトークンよりも能力がなければならぬ。この能力は新しいアプリケーションが獲得されるたび毎にC B S段階を実行するという事実のために必要である。ユーザの現存のバックアップはこの時無効にされて、追加的な実行権の発生を防止しなければならない。従つてこのトークン・データは代表的には頒布セットに使用できるトークン・データよりもはるかに長く、従つてこのトークンの1部が各C B Sトランザクションの1部として読取られ、この結果トークンの内容が変化し、トークンを完全に放出し尽す過程においてではなく途中で、前のバックアップ・ディスク・ファイルが無効になる。同じトークンがもし必要ならばI B Sトランザクションに使用できる。現在はこのようなバックアップ・セットのトークン上にシフト・レジスタ・メモリを使用せず、メモリはランダム・アクセス・メモリとして設計できていることを考慮している。しかしながら以下の説明では簡単のために、上記米国特許第92762

実行、たとえばユーザがその後のアプリケーションを得て、彼がこれをバックアップする権利を有する時に使用される。

コプロセッサは使用すべきトークン・データの長さに等しい長さの乱数を発生し、この乱数を使用してトークンを読取る、即ち照合して、トークンの応答を得る。コプロセッサはすでに平文のトークン・データ及びこれが発生した乱数が利用できるので正しい応答を予測即ち計算できる。トークンの計算された応答と実際の応答が比較される。もし比較がトークンの実際の応答と計算した応答が一致しないことを示すと、無効のトークンが提示されたことになり、誤りメッセージが戻され、このシーケンスは終る。しかしながら計算した応答と実際の応答が一致すると、トランザクションは継続する。バックアップ・トークンを読取る、即ち照合することによつて、その中に含まれるトークン・データが変更される（検査されるだけでなく）。トークン・データの変更によつて前のバックアップのすべてのコピーを無効にする。従つ

て9号と同じく、シフト・レジスタに基づくが、余分の長さのシフト・レジスタを含むものと仮定する。

C B S段階を開始する前に、ユーザはトークンとディスクより成るバックアップ・セットを入手している。ユーザが獲得したディスクは暗号化（された）トークン・データE C S K（トークン・データ）を記憶している。トークンは獲得した時は平文のトークン・データを記憶している。

C B S段階は次のように進行する。

ユーティリティ・プログラムがコプロセッサにC B Sシーケンスを開始することを知らせる。

コプロセッサはバックアップ・セットのトークン部分に対応する暗号化トークン・データを要求して、これを受取る。

コプロセッサはトークン・データを解読し、使用される分のトークン・データを選択する。トークン・データの一部の使用とはトークン・データのこの部分が破壊されることを意味する。トークン・データの残りの部分はC B S手順のその後の

でトークンに必要とされる長さについて云えば、トークンのビット長をC B S₆段階が実行される各度に使用される部分のビット長で割つた値の回数使用されるに過ぎない。トークンがもう1回トランザクションを行うのに十分なだけのデータを含む時は、このデータが読取られて破壊され、新しいバックアップ・トークンを開始できる。

このトランザクションの残りについては、コプロセッサ中で使用する暗号システムは任意の数字を有効なキーにすることができると仮定している。従つて乱数は有効なキーである。この性質はD E Sシステムの特徴である。

トークンの確認を検査した後、コプロセッサは第2の乱数でトークン・データの残りの未使用の部分、そのキー・メモリの許容部分及びそのソース・プロセッサを識別する一意的なスーパーバイザ・キー（U S K）を暗号化する。上述のように暗号化されたブロックは次にディスク上に記憶できる。このファイルはバックアップ設置手順のために使用できる。

上述の暗号化キーとして使用できる乱数のコピーはここで暗号化販売者キー手順を使用してスーパーバイザ・キーによつて暗号化される。このデータはここでバックアップ・ディスク上に記憶できる。

これ等のファイルは保護されるソフトウェアがコピーできるのと同じように、無限回コピーできる。しかしながら、このファイルはトークンとともにだけ使用でき、従つてトークンが途中で読取られない場合だけ使用できる。すでに説明したように、その後の転送動作はトークンを無効化できる。ユーザがCBS段階の最後の実行中に発生されたファイルの両方及び中間で読取られていないトークンにアクセスしたものと仮定して、バックアップ・セット設置段階を説明する。

さらに、中間に、ユーザのコプロセッサ、即ちソース・コプロセッサが故障したが、ユーザは他の(シンク)コプロセッサが利用できるものと仮定する。バックアップ・セット設置手順はシンク・コプロセッサ上にユーザのソース・コプロセ

販売者から獲得したディスクを使用することによつて除去する。

次にIBS段階は次のように進行する。

ユーティリティ・プログラムがシンク・コプロセッサにIBSシーケンスが始まろうとしていることを知らせる。最初コプロセッサにはバックアップ・ファイルを暗号化するのに使用する暗号化乱数キーを要求して、これが送られてくる。このキーは適切なCSKによつて解読される。次にコプロセッサは暗号化バックアップ・ファイルを要求して、それが与えられる。暗号化バックアップ・ファイルは暗号化したトークン・データ、(ソース)コプロセッサからのキー・メモリの内容及びソース・コプロセッサの一意的なスーパーバイザ・キーを含む。

(シンク)コプロセッサは前の段階で見出した乱数キーを使用してバックアップ・ファイルを解読し、トークン・データを取出し、トークンを検査する。トークン検査は頒布セットの検査と同じく進行する。即ち検査はトークンの内容を変更す

サ上に(バックアップの許可される程度に)存在した実行権のすべてを設置できる。しかしながら、ソフトウェアの販売者を保護するために実行権の全集りが許可期間によつて条件付けられる。許可期間中に、実行権の各々は動作できるので、ユーザは彼の(ソース)コプロセッサ中でこれ等を使用できる。許可期間中に、ユーザは彼の故障した(ソース)コプロセッサをハードウェア販売者に戻すことができる。ハードウェアの販売者は、コプロセッサが実際に故障したことを検査した後、(シンク)コプロセッサに(ソース)コプロセッサが故障しているので実行権についての許可期間の条件を上げることができることを確認させる暗号化したメッセージを有するディスクをユーザに与える。従つてIBS段階は2つのサブ段階を含む。第1のサブ段階は実行権を(シンク)コプロセッサに設置する。これ等の実行権を設置する時に、これ等は条件が付されて、許可期間が切れると無効になることを示す。IBS段階の第2のサブ段階は実行権についての条件をハードウェア

る。この結果、トークンはIBS手順を一回だけ遂行するのに使用できる。トークンが有効であると仮定すると、バックアップ実行権のセットが(シンク)コプロセッサの永久メモリ中に許可期間の範囲を決定するメ切りの日付とともに設置される。

ここで(シンク)コプロセッサによつて両コプロセッサの一意的なキーのコピーを含むメッセージが準備される。このメッセージは故障したソース・コプロセッサとともに戻され、ハードウェアの条件引上げメッセージはシンク・コプロセッサによつてシンク・コプロセッサに向けられたものであること、及び特定のソース・コプロセッサの実行権のセットのためのものであることが識別される。

この点で、(シンク)コプロセッサはすべての実行権がメ切りの日付について条件付けられた点を除き、(ソース)コプロセッサが最後にバックアップされた時と同じ条件にある。ユーザがハードウェア販売者から許可期間内に検査ディスクを

受取つたと仮定すると、I B S 段階の第2のサブ段階は次のように進行する。

ユーティリティ・プログラムは(シンク)コプロセッサにI B S シーケンスが完了しようとしていることを知らせる。(シンク)コプロセッサは検証ファイルを要求し、これを受取つてこれをスーパーバイザ・キー(シンクのU S K)によつて解読する。もしファイルが(ソース)コプロセッサを識別する一意的なキーを含むならば、条件が引上げられる。もしファイルが(ソース)コプロセッサの一意的なキーを含まない時は、誤りメッセージが戻され、サブ段階は完了する。

プロセッサ間通信によるバックアップ

前の章では、ハードウェア・バックアップはともに拡張トークンに依存するC B S 及びI B S 手順より成るものとしてハードウェア・バックアップを説明した。しかしながら、トークンもしくは拡張トークンの使用はハードウェア・バックアップにとつては不可欠でなく、中間のコプロセッサが先ずソース・コプロセッサと(C B S 手順のた

めに)通信し、次にシンク・コプロセッサと(I B S 手順のために)通信できるように配列できる限り、ハードウェア・バックアップの目的のために、トークンもしくは拡張トークンに代つて中間のコプロセッサが使用できる。

ハードウェア・バックアップ手順のプロセッサ間型の実形について説明する。

この説明の目的のために、ソース・コプロセッサは中間のコプロセッサと通信するものと仮定する。コプロセッサは最初送信するコプロセッサが選択したスーパーバイザ・キーによつて各々暗号化した暗号化乱数を交換する。既に説明したように、コプロセッサの各々は最初は使用された特定のスーパーバイザ・キーを知らないにもかかわらず、他のコプロセッサによつて転送されてきた暗号化乱数を解読して認識できる。受信するコプロセッサの各々は暗号化乱数を成功裡に解読したと仮定すると、これ等はこれ等の平文乱数を組合せてセッション・キーを発生し、その後の通信に使用する。

次にソース・コプロセッサはその永久メモリからアプリケーション・キー(A K)即ちバックアップされることが求められているキーを引出し、このキー情報及びそのU S Kをセッション・キーによつて暗号化して暗号化キー・ブロックを発生する。これはバックアップ・ディスク上に記憶される。セッション・キーは中間のコプロセッサによつて、このキーがバックアップ・セットに対応することを示す記述子とともにその安全なメモリ中に記憶される。この時点で、中間のコプロセッサはバックアップされつつあるA K(単もしくは複数)にアクセスするのに必要なキーを所有するが、このコプロセッサが信頼性のある受取り手でない場合は偽の実行権を発生できる。しかしながら、上述の識別化手順によつてソース・コプロセッサには中間コプロセッサが一族のメンバーであることが知らされている。そのようなファミリのメンバーとして、中間のコプロセッサの論理及び物理的安全性が転送されて来た実行権を偽の複製もしくは使用から防止する。実際の暗号化アプリケー

ションは通常的手段によつてI B S 手順のために中間のコプロセッサに送ることができる。

もしその後ソース・コプロセッサに中間のコプロセッサを介してバックアップされている実行権を転送するように要求されると、上述のようにソース・コプロセッサが中間のコプロセッサと通信してバックアップ権を無効にする迄はトランザクションは生じない。このような通信は3つの主要部分を有するだけである。第1の部分はすでに説明された識別子シーケンスである。これによつてソース及び中間のコプロセッサはこの通信の当事者が実際に一族のメンバーであるという条件が満足される。第2の部分はソース・コプロセッサからの中間のコプロセッサへの、バックアップ権のキーの無効を要求するメッセージの転送である。第3の部分は、自分がバックアップ権を記憶する真の中間のコプロセッサであり、権利が無効にされたことを示す中間のコプロセッサからの返答である。中間のコプロセッサ中に記憶されているバックアップ権はI B S 手順を使用して中間のコプロ

センサ中に設置できるか、もしくは異なる(シンク)コプロセッサ中に設置される。バックアップ権が中間のコプロセッサ中に設置される場合には、この動作は中間のコプロセッサに関連する上位計算機を介してユーザによつて容易に行われる。この権利はトークンを必要とすることなく設置できるが、すでに説明したように、これ等の権利は許可期間によつて条件が付されている。

中間のコプロセッサを使用するIBS手順が遂行される場合には、両コプロセッサは先ずセッション・キーを確立することによつてファミリのメンバーであるとして相互のアイデンティティを確立する。次に乱数キーがセッション・キーによつて暗号化され、シンク・コプロセッサに転送できる。シンク・コプロセッサがこのようにして許可され、許可期間を有するバックアップ実行権のセットが設置でき、上述のようにハードウェアの製造者のためのメッセージが準備される。

販売者のキーの暗号化

本明細書及び上記米国特許第927629号に

テキストを作成した、もしくは逆の解読に使用するキーを識別しようと試みる。この試みの困難さは、もちろん、選択した暗号システム及びバックトラック・タスクに利用できる計算能力による。ある暗号システム(DESのような)は非限定回数試みられる平文攻略に極めて抵抗性がある。

平文の攻略に対するコプロセッサの使用についての第1の制約では、コプロセッサはソフトウェアの販売者がハードウェアの製造業者からの許可を求めて実行権を発生するようになっている。この許可はハードウェアの販売者によつてソフトウェアの販売者に販売されるようになっている。従つて、平文の攻略に対するコプロセッサのソフトウェアの販売者についての1つの制約は、平文及び対応する暗号テキストのセットの発生に支払われる経済的コストである。他の技術は次の販売者キー(EVK)手順の使用である。

ユーティリティ・プログラムはコプロセッサにEVKシーケンスを開始しようとしていることを知らせる。コプロセッサは暗号化すべきキー(A

説明されているように、コプロセッサは又ソフトウェア解読キー(AK)を暗号化するというサービスをソフトウェアの販売者に与える。全ソフトウェアのコピー防止システムによつて与えられる保護にとつては、ソフトウェアの販売者の解読キーを暗号化するのに使用するスーパーバイザ・キー(CSK)が秘密に保持されることが重要である。ソフトウェアの販売者にコプロセッサの使用と無数のキーを暗号化する能力を与えると、ソフトウェアの販売者はハードウェアの販売者の暗号キー(CSK)に通常行われている選択された平文(を求める)侵害を試みることができる立場になる。ソフトウェアの販売者がハードウェアの販売者のキー(CSK)を知ることができると、そのソフトウェアの販売者はこれ等のキーを使用して他のソフトウェアの販売者の暗号化ソフトウェアを侵害することができる。選択された平文を攻略するには、侵害者が対応する平文及び暗号テキストのセットにアクセスすることが必要である。これ等セットを使用して、侵害者は平文から暗号

K)、代表的にはソフトウェア販売者のキーを要求して、これを受取る。キーはソフトウェアの販売者によつてスーパーバイザ・フラグの所望のセット値とともに与えられる。これ等のフラグはスーパーバイザが(たとえば)関連キーのバックアップもしくは転送を許可するかどうかを制御するものである。

アプリケーション・キーによつて検査されるか変更される(あるいはこの両方を行う)実行の条件も与えられる。次の説明でAKはこの過程による転送に準備された全データ(フラグ、キー及び条件)をさすものとする。

コプロセッサは乱数(RN)を発生しこれをキーAKの前端に埋込む。コプロセッサはキーの後端に認識フラグ(RF)を埋込む。上述のように、RFは正しいCSKが(解読中)に使用されたことを検査し、コプロセッサによつて正しい暗号システムが使用されたことを検査するのに適した任意のMACでよい。乱数及び認識フラグはユーザ、たとえばソフトウェアの販売者には知られ

ていない。コプロセッサは結果のブロックをスーパーバイザ・キー(CSK)によつて暗号化し、この結果をユーティリティ・プログラムに渡す。この結果、ECSK(RN, AK, RF)はCSKを知っている任意のコプロセッサによつて解読できる。ここで「.」はストリングの連結を示す。即ち01.111はストリング01111を生ずる。解読の結果のRN, AK, RFは3つの要素、乱数(RN)、解読キー(AK)及び認識フラッグ(RF)を含む。解読コプロセッサはサフィックスの認識フラッグの長さから先ずアクセスできる限り常にAKを識別できる。乱数の埋込みによつて、たとえ同じキー(AK)が多数回提示されたとしても、結果のブロック及び暗号化の結果が異なるので、平文の攻略を妨害することができる。EVK手順は解読コプロセッサが認識フラッグ(RF)を先に知っている限り、又すべての可能なCSKの先験知識を有する限り、特定のCSKの先見知識がなくとも暗号化ブロックを解読できるという他の利点を有する。さらに具体的に、各コ

プロセッサにはハードウェアの販売者によつてCSK1-CSK5が与えられているものとする。もし一号侵害を避けるための適切な手段をさらに与えなければ、使用される認識フラッグ(RF)は常に暗号キー自体であると仮定することができる。これ等の手段はブロック暗号化の場合はRNがブロックの長さの整数倍でないものとする。さらに暗号化に使用するコプロセッサは特定のソフトウェア・キーAKを暗号化するのにランダムにCSK3を選択したものとする。従つて暗号化ブロックはECSK3(RN, AK, CSK3)となる。CSK1-5にアクセスできる任意の他のコプロセッサはたとえキーCSK1-5のうちのどれが使用されたかを知らなくてもこのブロックを正しく解読できる。コプロセッサは唯キーCSK1-5の各々で暗号化ブロックを解読するだけでよい。唯1つの場合にだけ、平文版はサフィックスとして実際に使用する解読キーを含んでいる。

仮想キー・メモリ

本明細書及び上記米国特許出願第927629

号に説明されているソフトウェア保護機能の実際の具体化には、コプロセッサの実行権を記憶する能力にある制限を与えられなければならない。ユーザがコプロセッサ中に記憶できる数よりも多くの実行権を受取つた場合には、ユーザはめつたに使用しない実行権を転送トランザクションによつて移動可能な形に変換して戻すことができる。これ等の方法によるコプロセッサへのAKのスワップ・イン及びスワップ・アウトの動作はユーザの負担になることが容易に明らかであり、このような保護機構の精神に反する。しかしながら、コプロセッサにリソースを与えると、この問題を克服するのは容易である。

最も簡単な場合、AKはソフトウェアの一部のサービスにアクセスできる権利を表わす。この概念をわずかに拡張すると、メタAK(MAK)は実行権の集りをアクセスする権利を表わすものとなる。このキーの種類はすでに制限的な形でバックアップ・トランザクションに使用するRKもしくはランダム・キーに見られる。

ユーザがそのコプロセッサのすべてのキーの位置を充滿し、さらにもう一つを設置しようとしたとすると、コプロセッサはユーザにシステム・ディスク(ハード・ディスクもしくはフロッピー・ディスク)上にキーのファイルを開始させるオプションを与えることができる。この仮想キーの集りはMAKによつて暗号化形で記憶される。この集りがアクセスされてAKが移動もしくは加えられる度に、この集りは新しいランダムなMAKで再暗号化される。MAKはキー・メモリ中に記憶され、バックアップ及び転送を制御するフラグ中にMAKとしてマークされ、プロセッサによる正しい処理が行われるようにされる。アクセス毎にMAKの変更が必要であり、このようにしてコプロセッサによつて使用される仮想キーがユーザによつて所有される実行権の実際の集りと同期される。転送された仮想実行権を含む旧仮想キー・メモリのコピーはローディングしても正しく解読されないのでコプロセッサ上に紛れ込むことはない。

仮想キーを発生する(キー・メモリを解放する

目的のために) 特定の手順は次のように進行する。上位計算機上で実行中のユーティリティ・プログラムがコプロセッサに、ユーザが仮想キーの発生を要求したことを知らせる。コプロセッサはユーザに上位計算機を介して位置情報によつて仮想メモリ内に含まれるべきAKを識別することを要求する。ユーザの識別に基づいて、識別されたAKはキー・メモリからそのフラグ(転送、バックアップ等のための許可を識別する)、実行条件及び位置情報とともに取出される。コプロセッサは乱数を発生し、そのデータより成るブロックを暗号化する。結果の暗号化ブロックはここでディスク(ハードもしくはフロッピー)に書込まれる。乱数(仮想メモリのためのキー)が次に永久メモリ中に仮想化されたAKの1つに代つて書込まれる。仮想キーを乱数のMAKインデックスにマップするキー参照経路ファイルが平文で上位計算機のアクセス・ユーティリティに与えられ、参照のあいまいさが解決される。このファイルは乱数キーを、仮想化したAKを識別する位置情報と関連付ける。

デモンストレーション用ソフトウェア

仮想メモリ技術は又トークン・ソースもしくはカートリッジのような検証機構なしでデモンストレーション用ソフトウェアを支援することができる。トランザクションは以下説明するように進行する。キーの転送及びバックアップを許したり許さなかつたりするフラグのセット値はさらにキーの抹消を許したり許さなかつたりするフラグを含む。トークンがない状態でAKを受取らなければならない時には、このAKをローディングするためのデータ・ファイルは空のトークン記述子を含み、空のトークン・コネクタがトークンの照合に正しく応答する。この特殊なAKのためのフラグ・セットはバックアップを可能とするが、移動もしくは消去は可能としないセット値を含んでいる。このようなAKの設置には設置されたキー・メモリと仮想キー・メモリの両方を探索して、このキーが前に設置されたものであるかどうかを知る必要がある。

デモンストレーション用ソフトウェアの実行条

ユーザが後に仮想キーの1つによつて保護されているソフトウェアの実行を要求する。コプロセッサは先ず参照された位置のキーの使用を試みる。この位置にあるキーの正しさは通常のメッセージ許可技術によつてテストされる。このキーの検査は短かいメッセージ・ファイルに(保護されたプログラムの1部として与えられる)確認セクションをロードして解読することによつて遂行できる。もし解読によつて確認が有効であることがわかると、正しいキーが見出されている。もし確認が無効であると、アクセス・ユーティリティはもしユーザが実際にキーを有する時には、そのキーを含んでいる仮想キー・アクセス経路のリストを与えることができる。これ等のキーは仮想ブロックを解読し、適切なAKを引出すことによつて得られる。この時点で、コプロセッサは選択された、前もつて仮想化したAKをアクセスしている。

明らかに、仮想キー・メモリはそれ自身にMAKを含むことができ、キー・メモリの寸法は無限に拡張できる。

件には(好ましくは)経過時間もしくは使用回数を含む。上述のような条件の下で進行するAK設置はデモンストレーションの目的のためのソフトウェアの1部の実行権、代表的には1回使用の実行権の獲得を許容する。同じAKを設置しようとする2回目の試みは前の獲得を検出する。それはキー・メモリが探索されるが、キーが移動もしくは消去されていないからである。ユーザは無用のデモンストレーション用キーを仮想化することによつて、そのキー・メモリにこれ等の無用のキーで充満させることから守られる。ソフトウェアの販売者はデモンストレーション用ソフトウェアを使用することによつて彼のコードが繰返し、勝手に再設置されることから守られる。

E 実施例

E1 ソフトウェア資産保護機構

参照文献として、前記米国特許出願第927629号及び第927297号(トークンの実施例)があげられた。米国特許出願第927629号は

基本的なソフトウェア資産保護機構を開示している。この機構を簡単に第2図関連して説明する。ソフトウェア資産保護機構はユーザにある通信リンクもしくは経路14を介して接続した上位計算機10及びコプロセッサ20を含む複合計算システムの使用を要求する。米国特許出願第927629号に説明したように、経路14はたとえば上位計算機10とコプロセッサ20の両方を含むカバー内に封入された内部バスでもよく、コプロセッサ20内に含まれるI/O装置と上位計算機10に関連するI/O装置間のリンクでもよい。リンク14の特定の性質にかかわらず、コプロセッサ20にはユーザもしくは侵害者によるコプロセッサ20の内部への機械的干渉もしくはアクセスを防止する効果のある物理的安全性が与えられている。この物理的安全性は内部の破綻の境界によつて第2図に示されている。コプロセッサ20の2つの重要な機能部分は永久(持久)メモリ25及び一時メモリ26である。後者は通常の計算機の作業メモリ(RAM)と同種のものである。コ

プロセッサ20は永久メモリ25中に少なくとも1つの解読キーCSKが記憶された形でユーザに与えられる。解読キーCSKはコプロセッサ20の販売者によつて与えられ記憶される。ユーザが保護されたアプリケーションを実行するためには、このアプリケーションを実行する権利を永久メモリ25中に設置しなくてはならない。この実行権はソフトウェア解読キーAKによつて表わされる。上記米国特許第927629号に説明されているように、ユーザは実行権を設置するために、ソフトウェアの販売者からハードウェア・カートリッジ30及び頒布ディスク16を含む頒布セットを受取る。第2図に示したように、頒布ディスク16は3つのファイルを代表的には記憶している。1つのファイルは保護されたアプリケーションである。代表的な場合、保護アプリケーションは2つのサブ・ファイル中の2つの部分、即ち平文アプリケーション・ファイルAと暗号化された即ち保護された部分アプリケーション・ファイルBより成る。第2図に示すように、アプリケーション

・ファイルBはソフトウェア解読キーAKによつて暗号化されている。頒布ディスク16の第2のファイルはキーCSKによつて暗号化されたソフトウェア解読キーAKである。最後に頒布ディスク上の最後のファイルはソフトウェア解読キーAKによつて暗号化されたトークン・データ T_1 である。上記米国特許出願第927629号に開示されているように第3のファイルはかならずしも頒布ディスク16中に組込まれている必要はなく、ハードウェア・カートリッジ30中に組込まれてもよい。

ハードウェア・カートリッジ30は平文形でトークン・データ T_1 を記憶している。コプロセッサ20と同じように、ハードウェア・カートリッジ30は物理的安全性が与えられている。実行権を設置するために、ハードウェア・カートリッジ30は組合せ計算システムに結合即ちリンクされる。第2図は接続ケーブル18によるこの結合を表わす。コプロセッサ20は一時メモリ26中にある暗号化したソフトウェア解読キーAKを解読

する。ソフトウェア解読キーAKを永久メモリ25に受入れる前に、コプロセッサ20はハードウェア・カートリッジ30が偽造防止性のある照合/応答トランザクションで確認されたものであることが検査される(その内容 T_1 は暗号化ファイル $E_{AK}(T_1)$ に対応する)。ハードウェア・カートリッジ30は読取り後に破綻されるので、まだ使用されていない時はトークン・データ T_1 を含むだけである。ハードウェア・カートリッジ30が確認されたものであり、未使用であることを検査し(この過程でハードウェア・カートリッジの有用性を破綻した)後に、コプロセッサは実行権を受取り、その永久メモリ25中にソフトウェア解読キーAKを記憶する。ソフトウェア解読キーAKにアクセスすることによつて、保護されたアプリケーション・ファイルBが解読でき、コプロセッサ20の一時メモリ26中に記憶され、コプロセッサ20によつて実行できるようになる。物理的及び論理的的安全性のために、アプリケーション・ファイルBは実行過程中に平文形で一時メ

メモリ26中に記憶されるが、ユーザもしくは侵害者に利用されることはない。

E2 実行権の制約

上述のように、ソフトウェア資産保護機構はコプロセッサ20中に制約のない実行権を設置している。しかしながら本発明の特徴の1つは実行権に条件を付けることである。その例は期限の日付及び時間もしくは実行の回数である。第1図は第2図と似ているが、第1図ではアプリケーション・ファイルの保護部分が実行の規準、たとえば期限の日付もしくは時間が現在の日付けの前ならば実行を許可するという規準を含んでいる。第1図に示したように頒布セットを上位計算機10及びコプロセッサ20より成る複合処理システムに提示すると、実行権の設置は正確に第2図に関連して説明したのと同じように、コプロセッサ20は上位計算機10の仲介によつてトークン30の平文の内容(T_1)をディスク16から読取ったファイル $E_{AK}(T_1)$ の解説版とを比較することによつてトークン30を検査して(破棄する)。

を満足した場合にのみ実行が許可される。現在の日付もしくは時間はコプロセッサ20による要求によつて実行中のアプリケーションに供給される。

第4図は暗号化アプリケーション・キー・ファイル中に述べられている規準が許可される実行の残り回数を与えている点を除き第3図と類似している。実行権の設置により、ソフトウェア解説キーAKはカウンタCと関連し、アプリケーションの実行が要求されるたびに、カウンタCの内容が残された許可実行回数Cが0より大きいという規準に対してテストされる。次にカウンタCが1だけ減少される。Cが0より大きい限り実行は許可される。

制約が期限の日付であるか時間であるか、もしくは実行の回数であるかにかかわらず、コプロセッサ20には最初に定めた条件がもはや満足しなくなった時に関連ソフトウェア解説キーAKを抹消する命令を与えることが望ましい。従つてソフトウェア解説キーは永久メモリから自動的に除去され、実行権が削除される。

トークン30の確認及び未使用なことを検査した後、コプロセッサ20は永久メモリ25中にソフトウェア解説キーAKを読取る。実行の条件はAKと同じファイル中に記憶でき、AKと同時に設置される。第1図の場合には、コプロセッサ20は期限の日付、もしくは時間(あるいはこの両方)として解釈できるデータを記憶している。この解釈は任意の使用時にそのアプリケーションの保護部分によつて遂行できる。期限の日付メモリは第3図に示したようにソフトウェア解説キーAKと関連している。具体的に説明すると、第3図はソフトウェア解説キーAK及び期限の日付もしくは時間がコプロセッサ20の永久メモリ中に記憶されていて、トークン30が削除されている点を除き第1図と全く同じである。その後、保護アプリケーションがコプロセッサ20上で実行されるたびに、実行の許可の前に、アプリケーションは暗号化アプリケーション・ファイル中に現在の日付もしくは時間が期限の日付もしくは時間よりも遅くはないと述べてある規準を使用し、規準

E3 実行権の転送

第5図、第6図及び第7図は実行権の転送を説明するためのものである。第5図に示したように、コプロセッサ20はその永久メモリ25中にソフトウェア・キーAKの形で特定の保護アプリケーションを実行する権利を設置している。ユーザは実行権を転送するために、転送セットを得る。転送セットはトークン40及び付帯ディスク46を含む。トークン40はコプロセッサのハードウェア製造者のような信頼のおけるソースによつてトークン・データ T_2 がロードされていて、ディスクはハードウェアの販売者のキーによつて暗号化されたトークン・データが書込まれ、従つてディスクはファイル $E_{CSK}(T_2)$ を記憶している。ユーザは又第5図中に列挙されているファイルを含む原型のソース・ディスク16が利用できる。

第6図は転送シーケンスの第1の段階の後のこれ等の部品の条件を示す。さらに具体的には、転送セットのディスク46が読取られ、その内容が解説され、コプロセッサ20がその一時メモリ2

6中にトークン・データ T_2 を記憶できる。転送されるべきソフト解読キー AK は永久メモリ25から一時メモリ26に移動される。

転送シーケンスの次の段階はキー RK によつてトークン記述子 T_2 を暗号化して $E_{RK}(T_2)$ を求めること、ディスク46上に多数のファイル、即ちアプリケーション・ファイル A (平文)、暗号化アプリケーション・ファイル E_{AK} (アプリケーション・ファイル B)、ソフトウェア解読キー $E_{CSK}(AK)$ (必要ならば上述のようにして準備される)及び暗号化トークン・データ $E_{AK}(T_2)$ を書込むことである。従つて第7図に示した、2つの段階の終りには転送セットのディスク46はユーザがすでに取得していた原型のディスク16と略同じものになる。異なる点はディスク46と16上では(暗号形で存在する)トークン・データが異なること、(カウントのような)実行条件が変わった時に、新しい暗号化アプリケーション・キー・ファイル $E_{CSK}(AK)$ が使用されることである。このファイルは上述の同期

セットは異なるコプロセッサ上に保護アプリケーションの実行権を設置するのに使用できるが、実行権のための外部メモリの形として利用できる。たとえば、ユーザは望むときに実行権を、最初にこれを引出したコプロセッサ20に戻すことができる。

コプロセッサからコプロセッサへの実行権の転送を第19図及び第1表に示す。第19図はソース複合プロセッサが上位計算機10及びコプロセッサ20を含み、シンク複合プロセッサはシンク上位計算機110及びシンク・コプロセッサ120を含む。第19図に示したように、ソース・コプロセッサは転送さるべき実行権(AK)を含む。一般に両コプロセッサは同じスーパーバイザ・キーの集り(CSK)を含んでいるので、一般化するために、第19図はソース・コプロセッサは以下の過程でスーパーバイザ・キー CSK_1 を使用し、他方シンク・コプロセッサ120は異なるスーパーバイザ・キー CSK_2 を使用するものとする。どちらのコプロセッサも他方のコプロセッサ

機構によつて、新しいトークンに関連付けられる。従つて転送は侵害的手段によつて行なわれると無用のものになる。勿論コプロセッサ20上に実行権を設置する時には、ユーザはトークン・データ T_1 を含む彼のトークンが使用済であるので(従つて無効化している)、原型のディスク16上のファイル・セットを使用して他のコプロセッサ中に実行権を設置できない。しかしながら、ディスク46はユーザが現在所有するトークン40に対応する暗号化トークン・データを含んでいる。従つてユーザは保護されたアプリケーションを他のコプロセッサ上に保護アプリケーションを実行する権利を設置する立場にある。同時に前にソフトウェア解読キー AK を記憶していた元のコプロセッサはもはやこのキーを記憶していないことに注意されたい。従つて保護アプリケーションを実行する権利はコプロセッサ20からカートリッジ40及びディスク46を含む頒布セットによつて表わされたポータブルな形に移されている。

ディスク46及びカートリッジ40を含む転送

がどのスーパーバイザ・キーを使用するかをあらかじめ知っていないものとする。ソース及びシンク・コプロセッサは通信リンク200を介して相互接続されている。

上述のように、一度通信を確立すると、コプロセッサは識別子シーケンスを開始してそれ等自身が他方を一族のメンバであることを確かめる。この過程のあらましを述べた第1表の段階1では、各コプロセッサは乱数を発生し、従つてソース・コプロセッサは RN_1 をシンク・コプロセッサは RN_2 を発生する。段階2で乱数の各々は各コプロセッサによつて独立に選択されたスーパーバイザ・キーによつて暗号化され、暗号化した情報を転送して、段階3でシンク・コプロセッサ120は $E_{CSK_1}(RN_1)$ 、ソース・コプロセッサ20は $E_{CSK_2}(RN_2)$ がアクセスできるようになる。上述のように、ファミリのメンバだけが転送されて来た乱数を解読して認識できる。ソース及びシンク・コプロセッサは一族のメンバと仮定したので、各々は送信されたメッセージを解読

でき、従つてその後両コプロセッサはRN1及びRN2が利用可能になる。すでに説明したように、セッション・キーSKは両方の乱数の合成であるから、第1表の段階4で各コプロセッサは独立に同じセッション・キーSKを決定できる。

第 1 表

ソース・コプロセッサ	シンク・コプロセッサ
(1) RN1	RN2
(2) $E_{CSK1}(RN1) \rightarrow$	$\leftarrow E_{CSK2}(RN2)$
(3) $E_{CSK2}(RN2)$	$E_{CSK1}(RN1)$
(4) $SK = RN1 \cdot RN2$	$RN1 \cdot RN2 = SK$
(5) $E_{SK}(AK) \rightarrow$	$E_{SK}(AK)$
(6)	$\leftarrow E_{SK}(AK)$ の受取証
(7) AKの抹消	
(8) AKを抹消した \rightarrow	
(9)	AKをアクティベート

その後ユーザが転送さるべき実行権(AK)を識別した後、ソース・コプロセッサはセッション・キーSKによつて実行権を暗号化し、これをシ

ンク・コプロセッサに転送する(段階5)。この時点でシンク・コプロセッサ120は実行権AKにアクセスできるが、まだ実効はない。この実行性の欠如はコプロセッサの各々に課せられる信頼のおける安全手順によつて強請されるものである。シンク・コプロセッサはメッセージをソース・コプロセッサに戻し(段階6)、実行権の受取りを示す。これによつてソース・コプロセッサは実行権を抹消できる(段階7)。最後にソース・コプロセッサはメッセージをシンク・コプロセッサに転送し、実行権が抹消されたことを示す。この後初めて、段階9でシンク・コプロセッサは実行権をアクティベートできる。

その後ユーザが転送さるべき実行権(AK)を識別した後、ソース・コプロセッサはセッション・キーSKによつて実行権を暗号化し、これをシ

ンク・コプロセッサに転送する(段階5)。この時点でシンク・コプロセッサ120は実行権AKにアクセスできるが、まだ実効はない。この実行性の欠如はコプロセッサの各々に課せられる信頼のおける安全手順によつて強請されるものである。シンク・コプロセッサはメッセージをソース・コプロセッサに戻し(段階6)、実行権の受取りを示す。これによつてソース・コプロセッサは実行権を抹消できる(段階7)。最後にソース・コプロセッサはメッセージをシンク・コプロセッサに転送し、実行権が抹消されたことを示す。この後初めて、段階9でシンク・コプロセッサは実行権をアクティベートできる。

実行権もしくは実行権の集りの直接転送について説明したので、ネットワーク中の計算機間のライブラリ型のソフトウェア頒布のための安全な手順を説明することは簡単な拡張である。すでに説明した理由のために、保護ソフトウェアは上述のように各々上位計算機(たとえばPC)及びコプロセッサより成る複合計算システムに頒布できる

転送できる(ソフトウェア・フラグが転送を可能にすると仮定している)。実行権が転送される度に、ソースの権利の集りは転送された権利の分だけ減少し、権利が戻されるたびに、ソースの権利の集りはこの戻りによつて増加する。任意のソースの永久メモリは常に利用可能な権利のカウントを保持し、利用可能な権利の1つもしくはすべてが転送される度に内容が変更できる。ソースはたとえば、被転送ユーザが転送のために払出される口座をもつべきことを規定するように転送される権利を条件づけることができる。転送される権利は又使用時間もしくは使用回数等によつて条件付けることができる。ある場合、たとえば転送される権利に条件が付けられていて、将来のある日付に時間切れになつている場合には、ソースは前提としている転送トランザクションの日付が過ぎるとその実行権の集りをインクレメントできるようになつている。換言すると、転送できる権利が条件付けられていて、特定の日付に消滅するようになつているので、ソースにとつては

ンク・コプロセッサに転送する(段階5)。この時点でシンク・コプロセッサ120は実行権AKにアクセスできるが、まだ実効はない。この実行性の欠如はコプロセッサの各々に課せられる信頼のおける安全手順によつて強請されるものである。シンク・コプロセッサはメッセージをソース・コプロセッサに戻し(段階6)、実行権の受取りを示す。これによつてソース・コプロセッサは実行権を抹消できる(段階7)。最後にソース・コプロセッサはメッセージをシンク・コプロセッサに転送し、実行権が抹消されたことを示す。この後初めて、段階9でシンク・コプロセッサは実行権をアクティベートできる。

実行権もしくは実行権の集りの直接転送について説明したので、ネットワーク中の計算機間のライブラリ型のソフトウェア頒布のための安全な手順を説明することは簡単な拡張である。すでに説明した理由のために、保護ソフトウェアは上述のように各々上位計算機(たとえばPC)及びコプロセッサより成る複合計算システムに頒布できる

転送できる(ソフトウェア・フラグが転送を可能にすると仮定している)。実行権が転送される度に、ソースの権利の集りは転送された権利の分だけ減少し、権利が戻されるたびに、ソースの権利の集りはこの戻りによつて増加する。任意のソースの永久メモリは常に利用可能な権利のカウントを保持し、利用可能な権利の1つもしくはすべてが転送される度に内容が変更できる。ソースはたとえば、被転送ユーザが転送のために払出される口座をもつべきことを規定するように転送される権利を条件づけることができる。転送される権利は又使用時間もしくは使用回数等によつて条件付けることができる。ある場合、たとえば転送される権利に条件が付けられていて、将来のある日付に時間切れになつている場合には、ソースは前提としている転送トランザクションの日付が過ぎるとその実行権の集りをインクレメントできるようになつている。換言すると、転送できる権利が条件付けられていて、特定の日付に消滅するようになつているので、ソースにとつては

実際の転送が生じなかつた場合でも、その特定の日付の後に、前に転送した権利を再獲得することは適切である。ソースは又実行回数によつて測られる利用可能な権利のカウントを保持でき、勿論このようなソースはこのような権利の集りの全部もしくは一部を1回もしくは多数回のトランザクションで転送できる。又これ等の権利の一部は戻されてソースの集りを増すことができる。

任意の転送トランザクションで、コプロセッサからコプロセッサに実際に転送される唯一の必要なデータは暗号化キー（及び関連フラグ及び条件）であり、保護ソフトウェア自体、暗号化テキスト及び平文の部分とともに、通常の手段によつて転送できる。もし効果的ならば、すべての複合計算システムは保護ソフトウェア（暗号化形式）のすべて（もしくは一部）にアクセスできるようになつていて、キーだけを転送すればよいようにすることができる。転送しなければならないソフトウェアはキーが転送されるのと同じネットワークを介して転送できる。これに代つて、ソフトウェア

ソフトウェアが実行できなくなるが、実行能力は故障プロセッサを修理するか、代りのプロセッサを使用することによつて再び獲得できる。しかしながらコプロセッサ20の場合には、装置が故障すると、その中に記憶していた実行権が完全に永久に破壊される。従つて、ソフトウェアの販売者は顧客に実行権のハードウェアの形のバックアップ（予備）を与えることを望む。本明細書で説明されるハードウェア・バックアップは実行権の安全性に最小の影響を与えるように、たとえば偽造もしくは複製の実行権の回数及び持続時間を制限するようになつている。説明されるハードウェア・バックアップは実行権の転送の保留もしくは未完として概念的にとらえることができる。バックアップの必要性は将来の不確実な事件、たとえばコプロセッサ20の故障に依存するので、未完の実行権はコプロセッサ20と全く独立に実際の使用可能な実行権に変換できなくてはならない。この理由のために、複製実行権を発生できる部分が生じ、これが不法使用者によつて使用されてソフト

は郵便のような他のネットワークによつて転送できる。

E 4 実行権のバックアップ

以下の説明で、しばしばトークンの読取りについて言及するが、トークンを記憶するハードウェア・カートリッジの構造及びこれが読取られる方法については上記米国特許出願第927629号及び第927297号に説明されている。

全バックアップ・シーケンスは故障コプロセッサを前提とし、実行権を異なるコプロセッサ上に実行権を設置するので、故障コプロセッサをソース・コプロセッサと呼び、異なるコプロセッサをシンク・コプロセッサと呼ぶことにする。

ソフトウェア解説キーAKの形の実行権がすでに（ソース）コプロセッサ20中に記憶されているものとする。

コプロセッサ20は固有的な特徴を有するが少なくとも1点、即ち故障し得る点で任意の他のプロセッサと似ている。現在のソフトウェア頒布技術では、ユーザはプロセッサが故障した時に、ソ

ウェアの販売者によつて追求されている保護の裏をかかれる危険性を与える。しかしながら、以下説明するようにバックアップの設置は任意の（ソース）プロセッサにとつて一回しか遂行できず、許可期間（代表的な場合はバックアップの設置の時から測定する）に条件の付された、条件付き実行権を発生する。許可期間が終ると、さらにアクションがなされない場合には、バックアップ権から誘導できる実行権はもはや無効になる。この無法行為を著しく制限するこの代償は、実際のコプロセッサが故障した時に実行権の集りを再要求できる可能性が与えられないことである。合法的なユーザの場合は、ユーザはハードウェアの販売者のような信頼のおけるオブザーバに彼のコプロセッサが本当に故障したことを立証する責任がある。代表的な場合、この立証はハードウェアの販売者に故障したコプロセッサを物理的に送ることによつて行われる。故障したコプロセッサを検査して、ハードウェアの販売者はユーザに検査ディスクを与える。ユーザはこの検査ディスクは、（シンク）

コプロセッサに検証された許可を与え、バックアップ実行権に与えられた条件をなくし、実行権を、検査ディスクを獲得した時点に設定された条件によつてだけで条件付けられた実行権に変える。検査ディスクが置換えられた、元々実行権を記憶しているコプロセッサと同じコプロセッサの故障の証拠に基づいて準備されたことを保証するために、検査ディスクは暗号の形で故障したコプロセッサの一意的な識別子を帯びている。この一意的な識別子は条件の除去を可能ならしめるために必要である。シンク・プロセッサは又設置された実行権のセットから、条件を除去するためのメッセージがシンク・プロセッサによつて受取られることが意図されたものであることを認識できることが望まれる。この認識はメッセージ中に(シンク)コプロセッサの一意的キーのコピーを含ませて、そのメッセージの検証に使用するか、メッセージをシンク・コプロセッサの一意的なキー(USK)で暗号化することによつて達成される。ソース及びシンク・コプロセッサの識別子はともに、シン

ツ・セット設置(IBS)手順を実行する。従つて時刻 T_0 の後に、ユーザは故障したコプロセッサ20に代つてコプロセッサ120上で彼の実行権を使用できる。この期間中にユーザは故障したコプロセッサ20を販売者に送り、販売者はシンク・コプロセッサ120へのメッセージとして検査ディスクを作成する。許可期間が切れる前に、ユーザが検査ディスクを受取る限り、ユーザは検査ディスクを使用してバックアップ・セット設置手順を完了し(時刻 T_0 で)、コプロセッサ120中の実行権の条件付けを除去する。

第8図をここで参照するに、コプロセッサ20はソフトウェア解読キー AK_1 及び AK_2 によつて表わされた実行権の集りを有する。バックアップされる必要があるのはこれ等のキーである。バックアップ手順の第1の段階(第9図)はトークン50及びディスク56より成る未使用のバックアップ・セットをユーザが獲得する段階である。トークンはトークン・データ T_B を含み、 T_B は説明の目的のために、多くの部分、 $TB1$ 、 T

ク・コプロセッサによつて用意された暗号化(CSKによる)メッセージによつてハードウェアの販売者に利用可能になり、故障コプロセッサに戻る。

上述の手順は左から右に進む時間の関数として第17図に示されている。最初コプロセッサ20は実行権のセットを含んでいる。CBS手順(バックアップ・セットの作成)が開始され、これによつてカートリッジ及びディスクを含むバックアップ・セットが作成される。このバックアップ・セットは以下説明するように、転送セットと似ているが、CBS手順はユーザが新しい実行権を獲得するたびに逆行されて、単一のバックアップ・セットがコプロセッサのレパトリー内の各実行権のバックアップを与えるか、もしくはバックアップがソフトウェアの販売者によつて許可されている各アプリケーションにバックアップ権を与えることが好ましい。第17図は又、CBS手順の実行の後に、コプロセッサ20が故障したことを示している。その後ユーザは時刻 T_0 にバックア

B2・・・ TB_n より成るものとして示されている。ディスク56はキーCSKによつて暗号化したトークン・データを含む。ユーザはディスクの読取りを行わせ、上位計算機10上の適切なユーティリティ手順によつて、コプロセッサはディスク56から読取つた暗号化ファイルを解読して T_B をその一時メモリ26中に記憶する。ここで(第10図)、ユーザはトークン50を上位計算機10にカートリッジの接続ケーブル18を介して結合する。コプロセッサ20は使用するためのトークン・データの部分を選択する。ここでの説明の目的のためには、部分 $TB1$ を使用するように選択する。コプロセッサ20は乱数RNを発生し、CRと表わされる、 $TB1$ とRNの関数、即ちトークンの計算応答を計算する。コプロセッサ20は乱数RNをトークン50に印加して、最初の部分 $TB1$ を破壊的に読取る。トークン50のこの部分を読取ると、部分 $TB1$ が破壊され、コプロセッサ20には実際の応答が戻される。

第11図はこの状態における、即ちトークン5

0がもはやTB1を記憶せず、一時メモリ26が実際の応答ARを含んだ装置を示す。この時点でコプロセッサ20はARとCRを比較する。もしこれ等が一致しない時は、誤り条件が検出され、トークン50は確認されたものではないと考えられ、バックアップ手順は終了する。他方ARとCRが一致すると、コプロセッサはトークン50を確認されたものであると受取り、バックアップ手順をさらに続ける。第12図は上位計算機10が、コプロセッサ20によつて促されて、キーとして使用される新しい乱数(RK)を発生し、ディスク56に多数のファイルを含ませるように書込んだところを示している。最初に示されているファイルはトークン15の確認を検査するのに必要とされる読取り動作によつて修正された(たとえばトークン・データTBは暗号化前に部分TB1が削除されることによつて修正されている)単なるトークン・データの暗号化(RKによる)版である。コプロセッサ20によつてディスク56について形成される第2のファイルはトークン・デ

ータを暗号化するのに使用した乱数キーの暗号化版である。第3のファイルは第2のファイル中に暗号形で与えられたキーの下に暗号化されたAK及びその個々の関連フラグ及び条件のコピーである。このファイルは又ソース・プロセッサを識別する一意的なスーパーバイザ・キー(USKソース)のコピーを含んでいる。以下説明するように、検査過程中この後に、検査ディスクが販売者に証拠を与えた故障コプロセッサを識別する。検査過程は検査によつてソース及びシンク・コプロセッサの両方を正しく識別した時のみ、設置されている実行権のセットに課せられている満期条件を除去するように進行する。このファイル及びハードウェアのセットの、実行権獲得もしくは転送のトランザクションにおけるセットに対する構造上及び機能上の類似性は明らかであろう。ここでは、ユーザはトークン中の平文のトークン・データがディスク56中の暗号化トークン・データに対応するバックアップ・セット、即ちトークン50及びディスク56を有すると述べるだけで十分であ

る。これ等の装置はユーザに実行権AK₁及びAK₂を任意のコプロセッサ上に設置させるに十分である。それはこれ等の権利を設置するのに使用される手順がIBSであり、すなわち、実行権のセットは、導入されたとき、許可期間によつて条件づけられるからである。

第12図に示されたバックアップ・セットの発生に続き、ユーザが他のアプリケーションである、アプリケーション3の実行権を得て、コプロセッサ20中にその実行権AK₃を設置したとすると、彼はバックアップ・セットを使用して、AK₁及びAK₂だけでなく、AK₃をも包含するバックアップ・セットを発生することができる。この時、ユーザがディスク56を暗号化トークン・データを読取る複合計算システムに提示する。次にユーザは第9図乃至第12図に関連して説明した段階を遂行する。この処理の過程で、トークン部分TB2はこのトークンを検査する時に破壊され、その後、結果のトークン・データはTB3・・・TBnになり、ディスク56上の暗号化トークン記

述子ファイルもこれに対応し、ディスク56はさらにAK₃の暗号化版を含むようになる。もしユーザがこの点以前にディスク56のコピーを形成したとしても、これ等のディスクはCBSもしくはIBSトランザクションのために使用できない。それはこれ等のディスク上の暗号化トークン記述子及びRKがバックアップ・トークンを正しく検証し得ないからである。

勿論、ユーザは唯一つのトークン記述子部分がトークン50中に残される迄、トークン50及びディスク56より成るバックアップ・セットの使用を続けることができる。この時点で、他のCBSが遂行される場合には新しいバックアップが必要である。バックアップ・トークンの最後の部分がコプロセッサによつて読取られ、古いバックアップが無効にされ、新しいバックアップ・セットが開始される。

コプロセッサ20が故障した場合には、ユーザはバックアップ・セット設置(IBS)手順中にトークン50及びディスク56より成るバックアップ

・セットを使用できる。IBS手順については、第15図の参照から始めて以下に説明する。

第13図に示したように、ユーザはバックアップ・セットを上位計算機110及びコプロセッサ120より成る異なる複合計算システムに提示する。上位計算機110は原複合計算システム中の上位計算機10と異なる必要はなく、コプロセッサ120が故障したコプロセッサ20とは異なることだけが必要である。第13図に示したように、コプロセッサ120はその永久メモリ中に実行権を含んでいない。ユーザがディスク56を提示すると、暗号化ランダム・キー $E_{CSK}(RK)$ が読取られて解読され、ディスク56から暗号化トークン・データ $E_{RK}(TB2+TB3\cdots TBn)$ が読取られ、第14図に示したように解読される。ここで一時メモリ26はこのトークン記述子を平文形で含んでいる。次にコプロセッサ120は他の乱数SRNを発生して、平文のトークン・データの選択された部分(TB2)及び乱数SRNの関数CRを計算する。次にコプロセッサ120は

(USK)のコピー $E_{CSK}(USK)$ ソース及びUSKシンク)を含んでいる。この情報はハードウェアの販売者に提示される故障プロセッサがソース・プロセッサであることを実証し、シンク・プロセッサがソース・プロセッサの実行権のセットから期限の日付条件を解除するためだけの検査メッセージを作成するのに使用される。

コプロセッサ120がこの条件にある時は、ユーザは許可期間の継続中に AK_1 及び AK_2 によつてイネーブルされる保護アプリケーションを実行できる。許可期間はIBSの実行の開始時点から測定される。許可期間中にIBSを完了できない時は、実行権の行使は延期され、IBSを完了することによつて復位される。しかしながら、ユーザが販売者から適切な検査ディスクを受取ると、IBS手順は第15図に示したように完了する。第15図に示したように、ユーザは複合計算システムに検査ディスク66を提示する。検査ディスク66はその中にその故障が販売者に立証されたソース・コプロセッサを識別する一意的なスーパー

トークンを照合(質問)し、照合中に次の部分TB2を破壊して、実際の応答ARを発生する。第14図はこの時点での装置の状態を示す。次にコプロセッサ120はARとCRが一致するかどうかを判断する。もし一致しない時には、トークン50は確認されたものとみなされず、誤り条件に導入し、IBS手順が終了する。

次の段階(具体的には示されていない)で、 AK_1 、 AK_2 及びUSK(ソース)を含む暗号化ファイルがディスク56から読取られ、解読され、その個々の条件及びこれ等がすべて許可期間によつて条件付けられたという表示とともに永久メモリに記憶される。

ここでシンク・コプロセッサはハードウェアの販売者に与えるメッセージを用意する。このメッセージはスーパーバイザ・キーCSKによつて暗号化され、上位計算機によつてディスク上に記憶される。メッセージは共通のスーパーバイザ・キーCSKで暗号化されたソース及びシンク・コプロセッサの両方の一意的なスーパーバイザ・キー

ーバイザ・キーのコピーを帯びる、シンク・コプロセッサの一意的なキーによつて暗号化した単一ファイルを含む。コプロセッサ120は検査ディスク66を読取つて、ファイルを解読する。コプロセッサ120は次に検査ファイルの内容を記憶していたソースUSKのコピーと比較する。もしこれ等が一致しないと誤り条件が検査され、条件付き実行権は条件が付されたまま残る。これ等は許可期間の満了によつて中断する。他方、検査ファイルがすべての点で一致したものと仮定すると、コプロセッサ120は最終IBS段階を行うように許可される。これ等は実行権のセット AK_1 及び AK_2 の条件付けを抹消する。IBS手順が完了すると(第16図)、コプロセッサ120はその永久メモリ中に(ソース・プロセッサのバックアップ時の条件によつてのみ条件付けられた)実行権 AK_1 及び AK_2 を含む。この条件では、コプロセッサ120は故障する直前のコプロセッサ20と同じ条件にある(第8図)。従つて、上述の段階はコプロセッサ20のためのハードウェア

のバックアップを与えるが、ソフトウェアの販売者に最小のインパクトを与え、たとえば実行権の不許可の複製の可能性が最小になる。

トークンを使用するバックアップ手順はトークンとこれに関連する手順を使用し、セットが提示され、セットによつて表わされた実行権を受取ることができる任意のコプロセッサを検証する。バックアップの目的に中間のコプロセッサを使用する時は、トークンは不要である。安全性を与えるのはコプロセッサを支配する(ユーザの手の届く範囲外の)この手順である。従つてバックアップの目的のために中間のコプロセッサを使用する場合には、トークン・データを表わすディスク・ファイルが完全に不要なことを除き、トークン/ディスク対を使用する時に準備したのと実質的に同じようにしてディスク46が準備される。おそらくもう一つの他の差異は、バックアップ・セットを使用した時は、解読キーがスーパーバイザ・キーによつて暗号化されることである。このような構成は不用意に使用すると、このようなディスク

ション・ソフトウェアを実行することができる。この設定で、コプロセッサは1部もしくはそれ以上のソフトウェアを実行する権利を安全に記憶する機能及び上述のようにこのような実行権の処遇を定める機能を有する。しかしながら、上述のようにこれ等はコプロセッサの唯一の機能ではない。コプロセッサは又ソフトウェアの販売者によつて頒布可能パッケージを準備するのに使用される。ソフトウェアの販売者は任意の計算システムを使用して上記米国特許出願第927629号に説明したソフトウェア保護機能に従つて彼自身の秘密のキー(AK)によりトークン・データ及びソフトウェアを暗号化できるが、頒布可能なセットの他の部分はハードウェアの販売者のキーによつて暗号化されたソフトウェア解読キー、たとえばECSK(AK)である。ハードウェアの販売者のキーCSKがソフトウェアの販売者に知られないように、ソフトウェアの販売者はこの機能に(キーCSKが安全なメモリ中に与えられている)コプロセッサを使用できる。このサービスは、既

を与えられる任意のコプロセッサがバックアップを受取つてしまう。これを防止するために、解読キーを暗号化するキーはソース・コプロセッサと中間のコプロセッサ間で発生されるセッション・キーである。このようなセッション・キーの発生については既に説明された。中間のコプロセッサによつて記憶される唯一の情報は、セッション・キー及びこれがバックアップに関連することを示す表示である。転送手順は現在存在するバックアップの無効化を必要とするので、中間のコプロセッサはこれにセッション・キーが利用可能であるか、バックアップ・セット設置手順が有効であるかどうかを確認される。そして上述のように中間のコプロセッサがシンク・コプロセッサにセッション・キーを転送する限り、バックアップ・セットがシンク・コプロセッサ上に設置できる。

E5 販売者のキーの暗号化(EVK)

上述のように、多くのコプロセッサがユーザの複合計算システム(コプロセッサ以外に上位計算機を含む)中で使用でき、保護されたアプリケーション

に説明したように、キーCSK上での平文テキスト攻略をはかるために使用することができる。本発明では平文の攻略に特に抵抗力のあるDESを、この抵抗力を増強する以下説明する手順とともに使用することを考慮している。平文の侵害は侵害者に平文及び侵害しようとしているキーによつて暗号化された暗号テキストをアクセスすることを要求する。以下説明する手順を使用すると、侵害者はこのような情報にアクセスできなくなる。

第18図ソフトウェアの販売者が必要とするモードで使用されている代表的なコプロセッサ220を示す。ここで入力1乃至それ以上のソフトウェア解読キーAK₁、AK₂・・・等であり、出力はキーCSKによつて暗号化したこれ等のキーである。すでに確立した約束により、コプロセッサ220は物理的及び論理的に安全にされている。この安全性は破線の内部境界によつて示されている。コプロセッサ220が各ソフトウェア解読キーの入力に対して単に出力ECSK(AK)を出力するだけならば侵害者には選択された平文

攻略に必要なデータが容易に与えられる。

本発明のこの態様に従えば、コプロセッサ20は暗号化の前に解読キーを正する。しかしながらこの修正はすべての他のコプロセッサが知っているものであり、従つてすべての他のコプロセッサは逆修正をほとんどすることができる。一般的に説明すると、各ソフトウェア解読キー AK_1 はその前後に埋込みを行うことによつて修正される。さらに具体的には、第18図に示されたように、代表的には既知のビット長のメッセージ確認コード(MAC)の形のサブフィックス認識フラグ(RF)がプレフィックス乱数(RN)とともに使用される。従つてコプロセッサ220は AK_1 の提示に応答して、 $RN_1 \cdot AK_1 \cdot RF$ (ここで「 \cdot 」は連結を示す)。コプロセッサはここでCSKのセットの中のあるキーCSKによつて結果のデータ・ブロックの連結を暗号化して $E_{CSK_1}(RN_1 \cdot AK_1 \cdot RF)$ を発生する。DESを理解すると、CSK₁にアクセスした他のコプロセッサはこの結果を解読してストリング $RN_1 AK_1 RF$ を発

$E_{CSK_k}(RN_3 \cdot AK_3 \cdot RF)$ 等を識別できる。この明文及び暗号化データの組はこれ等の種々のストリングを暗号化したキーを識別しようとする試みる侵害者にとっては、明文 AK_1 、 AK_2 、 AK_3 等を知っていたとしてもほとんど助けにはならない。

従つてソフトウェアの販売者が使用してそのソフトウェア解読キーを暗号化するためのすべてのコプロセッサを要するコプロセッサ220は次の販売者キー暗号化(EVK)手順を遂行する。ユーティリティ・プログラムはコプロセッサ220にEVKシーケンスを開始する事を知らせる。コプロセッサ220は乱数(RN)を発生しこれをキーAKの前端に埋込むためのプリフィックスとして使用する。コプロセッサ220は又認識フラグ(RF)をサブフィックスとして使用して後端に埋込む。結果のブロック即ちストリングはスーパーバイザ・キーCSKの下に暗号化され、結果が上位計算機に受渡される。

このEVK手順は認識フラグを適切に選択す

生できることは明らかである。解読を行うコプロセッサはRFの内容及びビット長のみならず AK_1 のようなソフトウェア解読キーのビット長についての知識を前もつて有するので、コプロセッサはメッセージ $E_{CSK_1}(RN_1 \cdot AK_1 \cdot RF)$ をCSKのセットの各々でメッセージ $E_{CSK_1}(RN_1 \cdot AK_1 \cdot RF)$ を解読し、最後にRFを正しく解読できる。一度コプロセッサが正しいCSKを発見して、暗号化情報を解読すると、 AK_1 を分離して、これを特に識別できる。プリフィックス RN_1 は(前に述べられたように、ある他の検査タスクを実行することが必要でないなら)単に破棄される。

DESを理解すると、明文の侵害は AK_1 、 $E(AK_1)$; AK_2 、 $E(AK_2)$; AK_3 、 $E(AK_3)$ 等の明文及び暗号化情報のセットの知識を必要とすることがわかるが、コプロセッサ220にアクセスできる者は容易に次の組、 AK_1 、 $E_{CSK_1}(RN_1 \cdot AK_1 \cdot RF)$; AK_2 、 $E_{CSK_2}(RN_2 \cdot AK_2 \cdot RF)$; AK_3 、

ることによつて、データ・ブロックは1つのコプロセッサによつて選択されるスーパーバイザ・キーによつて暗号化される。暗号化ブロックは解読するコプロセッサがすべての可能なスーパーバイザ・キーを含むスーパーバイザ・キーのセットにアクセスできる限り、どのスーパーバイザ・キーが暗号化するのに使用されたかをあらかじめ知らなくても、解読できる。この解読は両方のプロセッサ中の予備知識、メッセージ許可コードを含む多くの手段並びに認識フラグ(RF)として暗号化スーパーバイザ・キーを選択することによつて達成できる。たとえば、暗号化コプロセッサが AK_3 を暗号化する目的のためにCSK₃を選択したと仮定する。上述の手段によつて暗号化するコプロセッサは $E_{CSK_3}(RN \cdot AK_n \cdot RF)$ を発生する。解読するコプロセッサはどのCSKが暗号化に選択されたかを知らないが、すべてのCSKにアクセスできるものと仮定する。解読するコプロセッサはCSK₁で始まつて各解読キーで順番に暗号化ブロックを解読する。ブロックを

解説するたびに、サフィックスが解説キーと比較される。サフィックスと解説キーが一致すると、解説するコプロセッサは暗号キーを識別し、同時に AK_n にアクセスできる。それはこのブロックが正しい解説キーを使用してすでに解説されているからである。同じように、もし予備知識を使用すると、予想したストリングは RF 位置に見出されなくてはならない。もしくは MAC を使用すると、 MAC は AK_n もしくは RN, AK_n に対して有効でなければならない。

上述の説明では、参照はソフトウェア解説キー AK についてなされた。以下に説明する理由で、 AK と呼ばれるブロックはソフトウェア解説キーだけでなく、いくつかのフラグを含んでいる。その条件はソフトウェアの販売者によつて選択され、ある手順の実行を許容したりしなかつたりする。たとえば、1ビットのフラグをセットして、ハードウェア・バックアップ手順の使用を許容し、許容しなかつたりする。ハードウェア・バックアップ・フラグがバックアップ手順を許容しないよう

にセットされている場合には、解説キーを記憶する任意のコプロセッサはバックアップ手順を制限してこのキー（及びそのようにマークされた他のキー）を除去する。ソフトウェアの販売者は一度設置したソフトウェア解説キーの転送を禁止したい場合がある。この目的のために、1ビット転送フラグが与えられて、転送動作を許容したり許容しなかつたりするのに使用できる。上位計算機が転送手順を要求すると、この手順は転送フラグが転送動作を可能ならしめるようにセットされている時のみ許容できる。これ等の制限の各々は各コプロセッサ中に与えられた機能に与えられるデータとして理解される。 AK は又個々のアプリケーションによつてテストされる上記の実行条件も含んでいる。

第2表はしばらく使用されており、実行権の集りを記憶する代表的なコプロセッサの永久メモリ25の一部の内容を示す。

第 2 表

2進フラグ					多重バイト記入項目		
メ タ	条 件	抹 消	転 送	バック アップ	条 件	位置及び 検査情報	キ ー
0	0	0	0	0	—		CSK_1
0	0	0	0	0	—		CSK_2
⋮							⋮
0	0	0	0	0	—		CSK_N
0	0	1	0	1	—		AK_1
0	0	0	1	1	—		AK_2
0	1	1	1	0	データ		AK_3
⋮							⋮
0	1	1	1	0	データ		AK_N
1	0	1	1	1	—		MAK_1
1	0	1	1	1	—		MAK_2

第2表に示したように、永久メモリ25は複数の異なるキーの各々のためのレコード（即ち記入項目）を含む。 CSK_1 乃至 CSK_n 、 AK_1 乃至 AK_n 及び MAK_1 及び MAK_2 が第2表に示されている。各レコードは多くのフィールドを含む。フィールドの1つはキー自体である。各キーには多数の2進フラグが関連し、フラグ：メタ、条件、抹消、転送及びバックアップの各々に1ビットが存在する。このリストはこのようなフラグの有用な部分集合であり、この集合はこの分野の専門家によりほとんど確実に拡張できることは明らかである。2進フラグはたとえば第1の欄の下に、キーがメタ・キーであるかどうかを示している。第2表では最後の2つのレコードだけがメタ・キーの記憶を示している。次の欄（条件と題する）中の2進1はキーが条件付けられていることを示す。第2表ではキー AK_3 及び AK_n が条件付けられている。第3の欄（抹消と題する）は2進0によつてキーの抹消が許容されていないことを示す。この条件はスーパーバイザの各々及び特

定のアプリケーション・キー A K₂ に適用できる。第4欄(転送と題する)は2進1によつて A K₁ を除くすべてのキーが転送が許可されていることを示している(ただしスーパーバイザ・キーは除く。このキーの転送は全く必要でなく、又望ましくない)。第2表の2進フラグ部分の下の最後の欄はバックアップと題し、2進1は関連キーがバックアップが許可されていることを示す。第2表に示したように、キー A K₃ 及び A K₀ はバックアップは許可されない。

第2表に示したメモリはキーの各々について多くの多重バイト記入項目を有する。多重バイトの記入項目の1つは条件と題し、このフィールドは条件の付せられたキーに関連するデータを含む。従つてキー A K₃ 及び A K₀ はこのフィールドに、これ等のキーが解説するアプリケーション中に記憶された規準によつてテストされ、実行が許可されているかどうかを判断するデータを含んでいる。第2表の多重バイト記入項目の最後の欄はアプリケーション・キーの探索を助け、探索した時にそ

のキーの検査を助ける位置及び検査情報を与える。

E6 ソフトウェアの返品

上述の説明から、実行権のための処遇について説明した手順はソフトウェア販売者に、彼の顧客だけでなく彼自身にとつても公平なソフトウェア返品対策を与える。経済的な理由のために、ある販売者は本発明には関連ないとはいえ、ソフトウェアの返品をある一定の期間(一種の保証期間)内に制限したいであろう。ソフトウェアの販売者はたとえばユーザに特定のソフトウェアのアプリケーション・パッケージを実行する権利を含む転送セットを販売者に与えることを要求することによつてソフトウェアの返品を(販売者の選択による全面もしくは部分的信用で)認めている。ユーザがこのような転送セットを作成する方法は既に説明した。もしユーザがソフトウェアの販売者に返品を求める特定のアプリケーション・パッケージを含む有効な転送セットを提示すると、ソフトウェアの販売者には(ソフトウェアのコピー防止機構の動作によつて)ユーザ自身がもはやこのソ

フトウェアを実行する権利を保持しないことが保証される。上述の如く、転送セットの発生はコプロセッサから解説キーの削除を必要とする。

上述のことから、本発明は上述の米国特許出願第927629号中に説明したソフトウェア保護機構を実行権の処遇に広い柔軟性を与えることは明らかである。本発明はこれ等の手順をいくつかの論理動作及びそれ等の相互関係によつて説明した。本明細書中の説明からこれ等の論理動作を実行させるソフトウェアを作成することはこの分野の専門家にとつて明らかであろう。従つて上述の手順を具体化するソフトウェアの特定の説明は不要であろう。

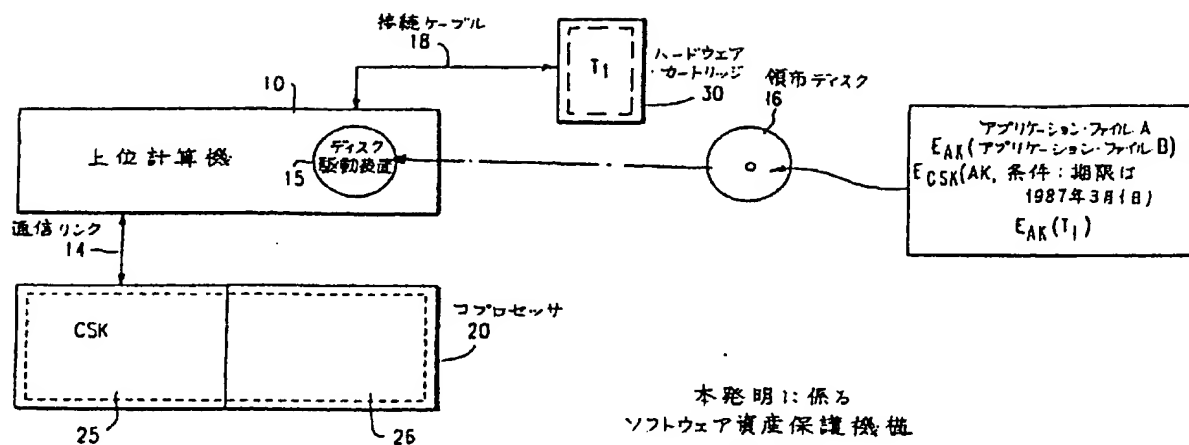
F 発明の効果

本発明に従い、コプロセッサ中に存在するソフトウェアの実行権に条件を付け、その処遇を定め、これを転送する方法が与えられる。

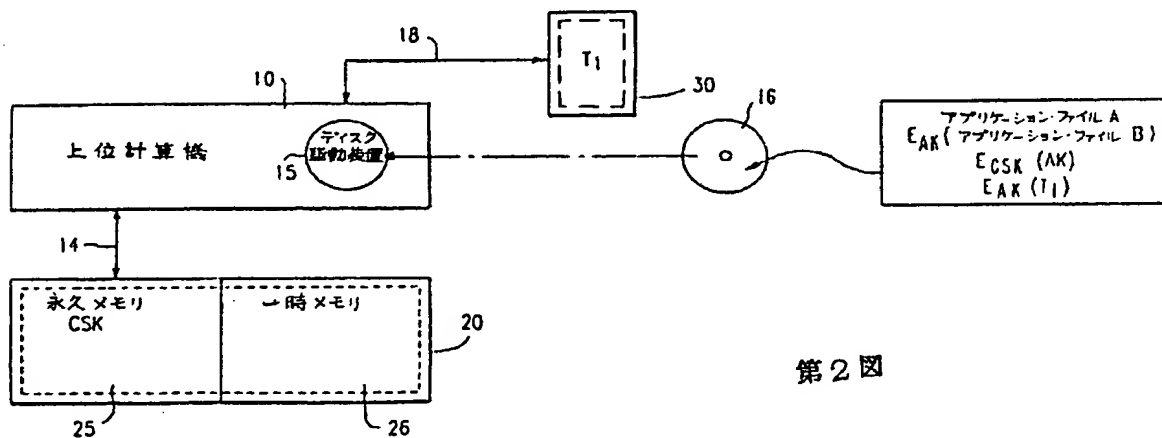
4. 図面の簡単な説明

第1図、第3図、第4図は第2図と類似であるが、本発明に従う、条件の付された、ソフトウェアの実行権の用途及び応用を説明する図である。第2図はソフトウェア資産保護機構の主要部品及びその相互の関連を示す図である。第5図、第6図及び第7図は実行権の転送を説明する図である。第8図、第9図、第10図、第11図、第12図、第13図、第14図、第15図、第16図はバックアップ手順の2段階(CBS及びIBS)を説明する図である。第17図はCBS及びIBS手順に使用する代表的シーケンスを示す図である。第18図は販売者キーの暗号化を説明する図である。第19図は実行権の直接転送を説明する図である。

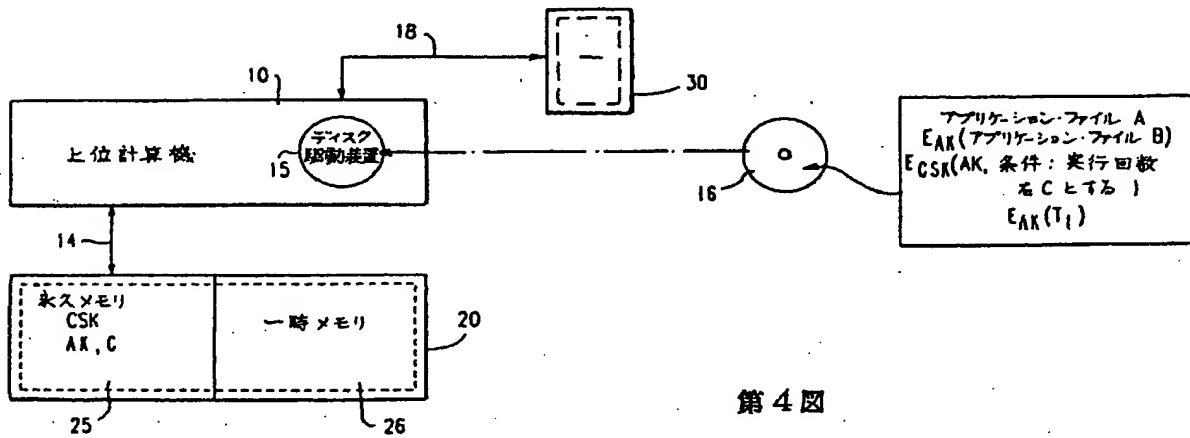
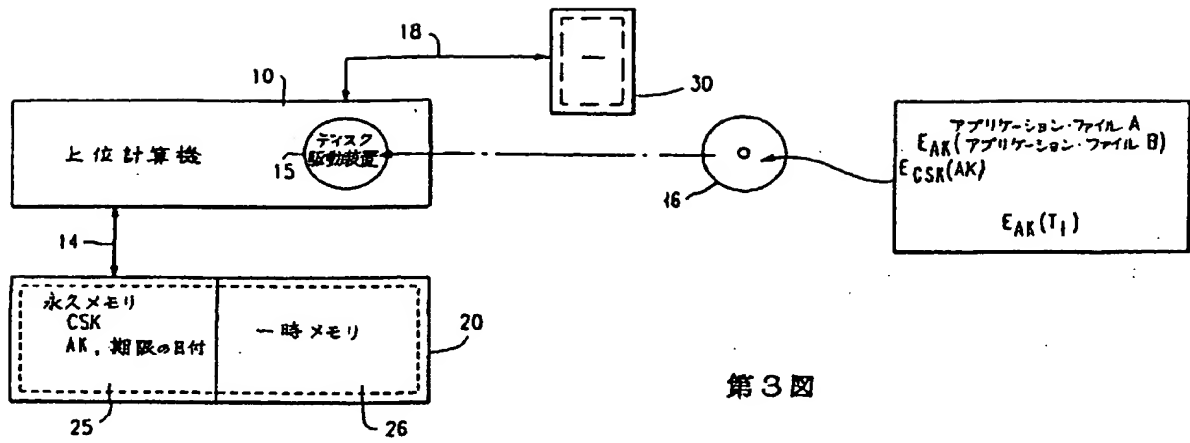
10...上位計算機、14...通信リンク、15...ディスク駆動装置、16...頒布ディスク、18...接続ケーブル、20...コプロセッサ、25...永久メモリ、26...一時メモリ、30...ハードウェア・カートリッジ。

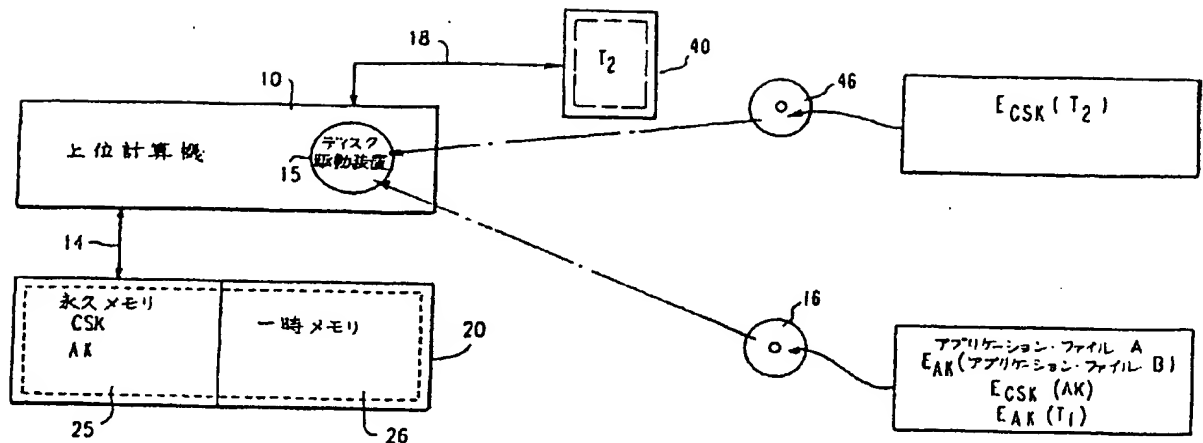


第1図

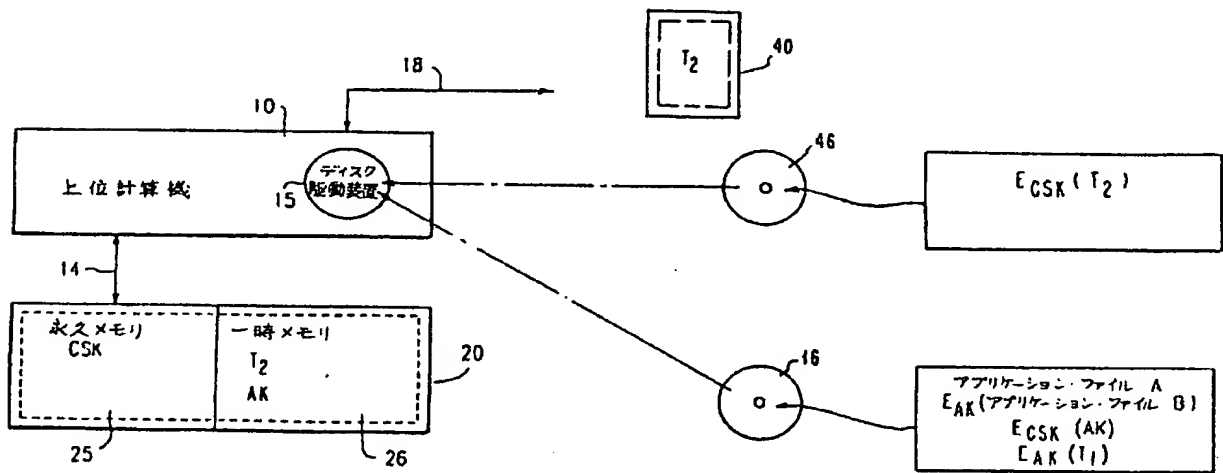


第2図

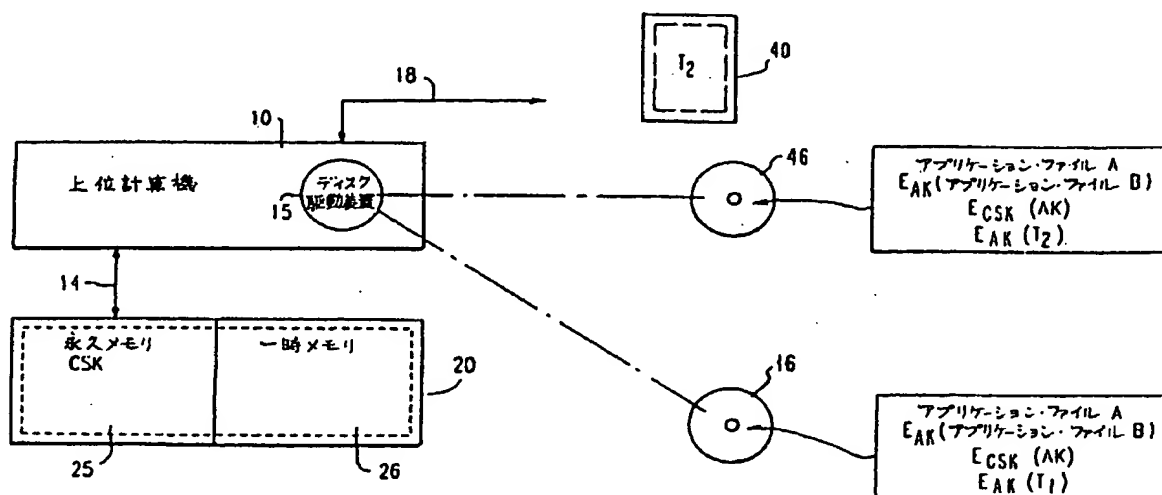




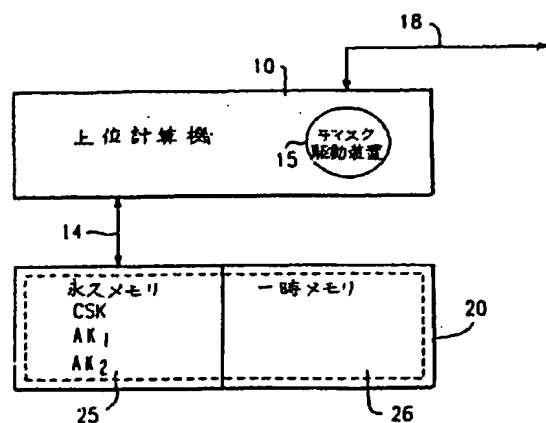
第5図



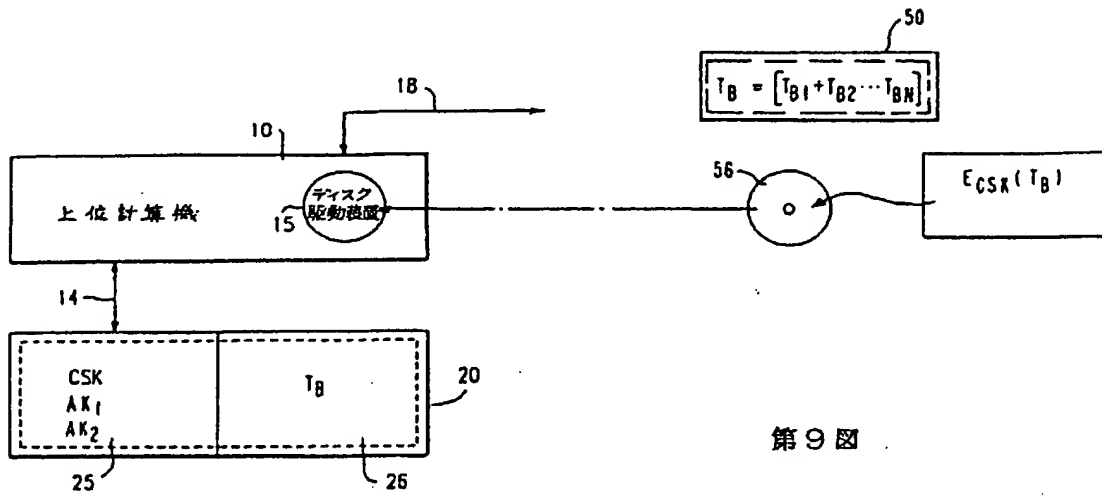
第6図



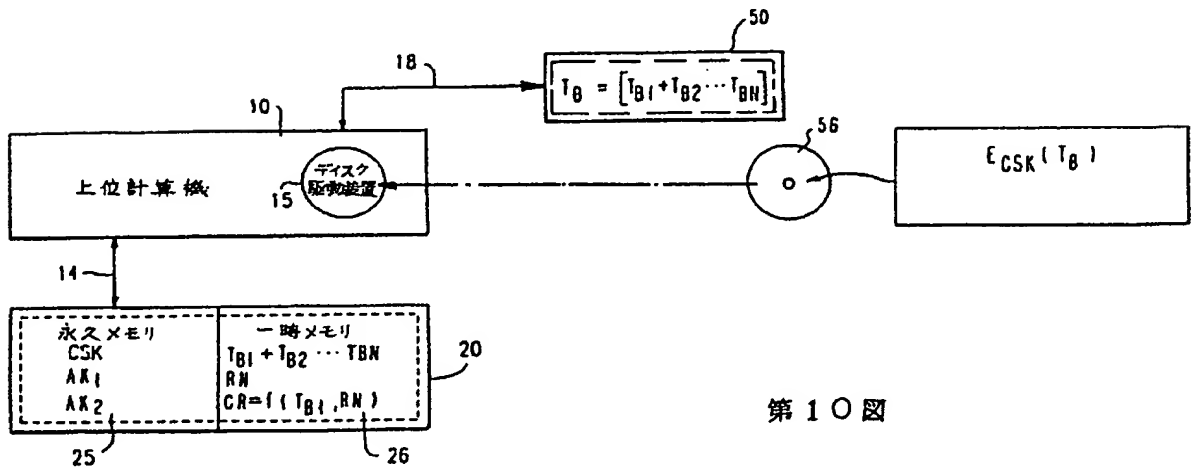
第7図



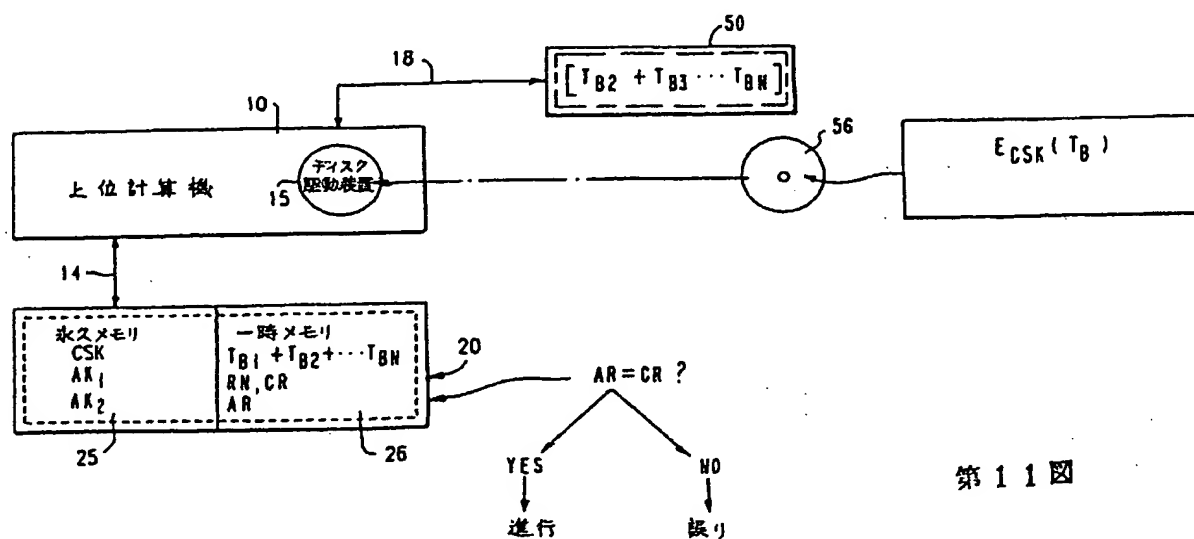
第8図



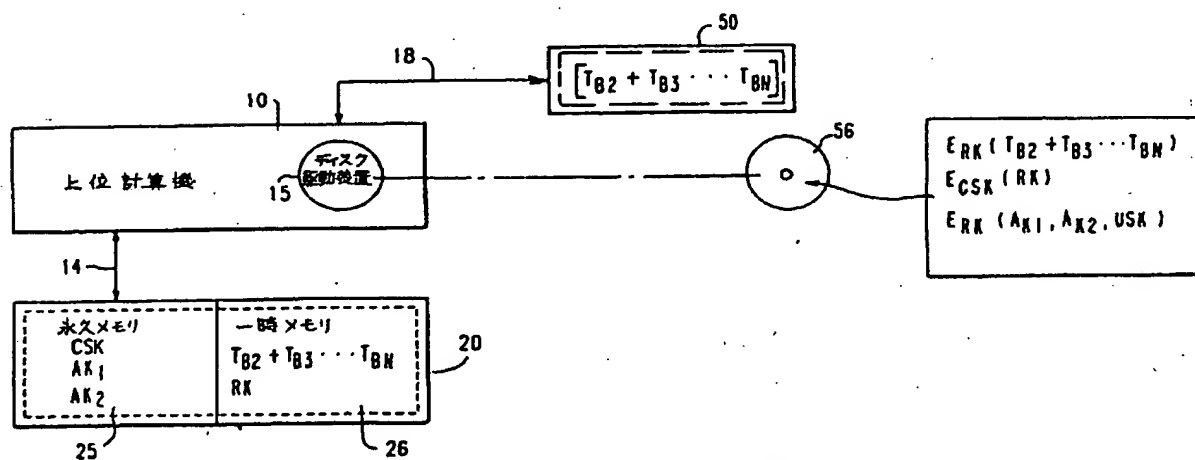
第9図



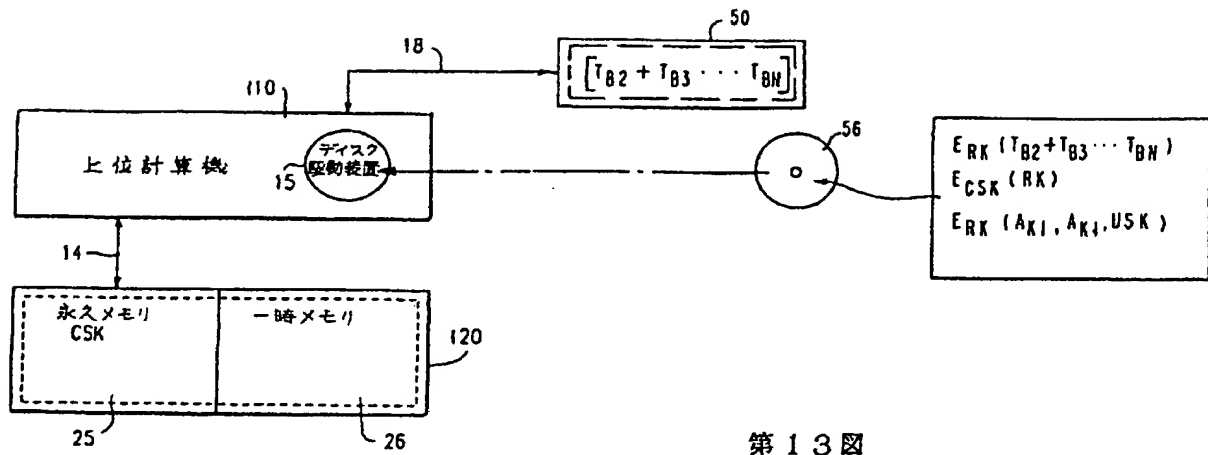
第10図



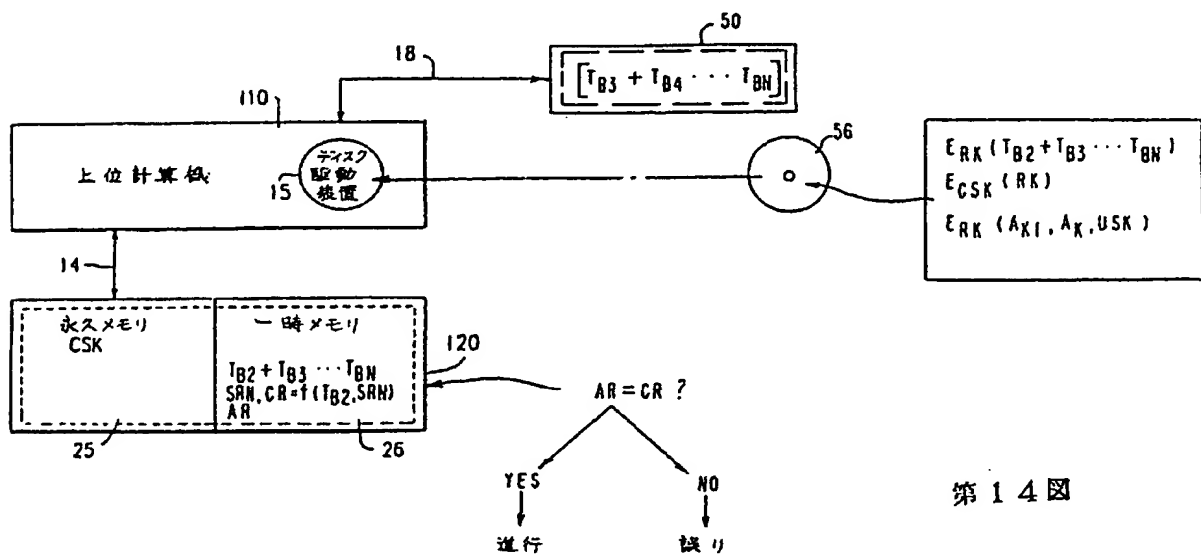
第11図



第12図

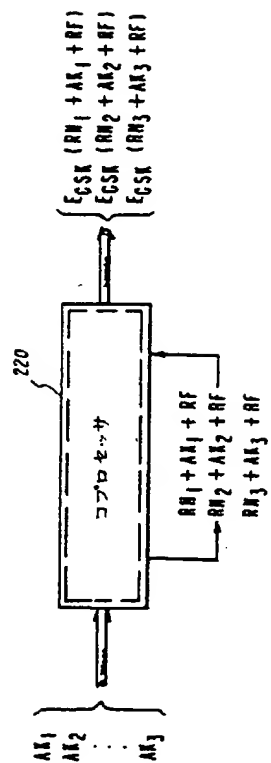


第13図



第14図

第18図



第19図

